

**Конспект лекций по дисциплине
“ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ”**

СОДЕРЖАНИЕ

Раздел 1. ПОНЯТИЕ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ	3
1.1 Содержание информационной технологии	3
1.1.1 Определение информационной технологии	3
1.1.2 Инструментарий информационной технологии	4
1.1.3 Информационная технология и информационная система	5
1.2 Этапы развития информационных технологий	6
1.3 Особенности новых информационных технологий	8
1.4 Проблемы использования информационных технологий	9
Раздел 2. ВИДЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ	10
2.1 Общая классификация видов информационных технологий	
Ошибка! Закладка не определена.	
2.1.1 Структура управления организацией	Ошибка! Закладка не определена.
2.1.2 Классификация видов информационных технологий	10
2.2 Информационная технология обработки данных	14
2.3 Информационная технология управления	16
2.4 Автоматизация офисной деятельности	19
2.5 Информационная технология поддержки принятия решений	23
2.6 Экспертные системы	27
2.6.1 Типы экспертных систем	28
2.6.2 Виды знаний	29
2.6.3 Способы формализованного представления знаний в БЗ	29
2.6.4 Области применения ЭС	31
Раздел 3. ОРГАНИЗАЦИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ	34
3.1 Модели информационных процессов передачи, обработки, накопления данных	34
3.1.1 Обобщенная схема технологического процесса обработки информации	34
3.1.2 Сбор и регистрация информации	35
3.1.3 Передача информации	38
3.1.4 Обработка информации	39
3.1.5 Хранение и накопление информации	41
3.2 Системный подход к решению функциональных задач и к организации информационных процессов	43
Раздел 4. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В РАЗЛИЧНЫХ ОБЛАСТЯХ ДЕЯТЕЛЬНОСТИ	46

4.1	Информационные технологии в системах организационного управления	46
5.3.1	ЭВМ при выборе решений в области технологии, организации, планирования и управления производством	46
5.3.2	Возможности использования новых информационных технологий в системах организационного управления	47
4.2	Информационные технологии в обучении	50
4.3	Автоматизированные системы научных исследований	52
4.4	Системы автоматизированного проектирования	53
4.5	Геоинформационные системы и технологии	54

Раздел 5. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ **57**

5.1	Технологии распределенных вычислений (РВ)	57
5.2	Распределенные базы данных	59
5.3	Технологии и модели "Клиент-сервер"	61
5.3.1	Модель файлового сервера	62
5.3.2	Модель удаленного доступа к данным	63
5.3.3	Модель сервера базы данных	65
5.3.4	Модель сервера приложений	66
5.4	Технологии объектного связывания данных	67
5.5	Технологии реплицирования данных	70

Раздел 6. ТЕХНОЛОГИИ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ **73**

6.1	Понятие о компьютерном математическом моделировании.	73
6.1.1	Общие сведения о компьютерном математическом моделировании	73
6.1.2	Классификация математических моделей	73
6.2	Этапы, цели и средства компьютерного математического моделирования	75
6.2.1	Моделирования случайных процессов	78
6.2.2	Особенности имитационного моделирования производственных систем	79

Раздел 7. ТЕХНОЛОГИИ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ **81**

7.1	Общая характеристика технологии создания программного обеспечения	81
7.2	Современные методы и средства разработки программного обеспечения	85
7.1.1	Современные методы разработки ПО	85
7.1.2	Инструментарий технологии программирования	89
7.1.3	Средства для создания приложений	89
7.1.4	CASE-технологии	92
7.3	Языки и системы программирования	93
7.3.1	Развитие языков программирования	93

7.3.2	Современные системы программирования	97
7.4	Архитектура программных систем	99
Раздел 1. ПОНЯТИЕ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ		

1.1 Содержание информационной технологии

1.1.1 Определение информационной технологии

Технология при переводе с греческого (techne) означает искусство, мастерство, умение, а это не что иное, как процессы.

Под процессом следует понимать определенную совокупность действий, направленных на достижение поставленной цели. Процесс должен определяться выбранной человеком стратегией и реализоваться с помощью совокупности различных средств и методов.

Под технологией материального производства понимают совокупность средств и методов обработки, изготовления, изменения состояния, свойств, формы сырья или материала. Технология изменяет качество или первоначальное состояние материи в целях получения продукта.

Информация является одним из ценнейших ресурсов общества, наряду с такими традиционными материальными видами ресурсов, как нефть, газ, полезные ископаемые и др., а значит, процесс ее переработки по аналогии с процессами переработки материальных ресурсов можно воспринимать как технологию. Тогда справедливо следующее определение.

Информационная технология (ИТ) - совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта).

Цель информационной технологии - производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия.

Практическое приложение методов и средств обработки данных может быть различным, поэтому целесообразно выделить глобальную базовые и конкретные информационные технологии.

Глобальная информационная технология включает модели методы и средства, формализующие и позволяющие использовать информационные ресурсы общества.

Базовая информационная технология предназначена для определенной области применения (производство, научные исследования, обучение и т.д.).

Конкретные информационные технологии реализуют обработку данных при решении функциональных задач пользователей (например, задачи учета, планирования, анализа).

Как и все технологии, информационные технологии находятся в постоянном развитии и совершенствовании. Этому способствуют появление

новых технических средств, разработка новых концепции, методов организации данных, их передачи, хранения и обработки, форм взаимодействия пользователей с техническими и другими компонентами информационно-вычислительных систем.

Расширение круга лиц, имеющих доступ к информационно-вычислительным ресурсам систем обработки данных, а также использование вычислительных сетей, объединяющих территориально удаленных друг от друга пользователей, особо остро ставят проблему обеспечения надежности данных и защиты их от несанкционированного доступа. В связи с этим современные информационные технологии базируются на концепции использования специальных аппаратных и программных средств, обеспечивающих защиту информации

Следующим шагом в совершенствовании информационных технологий, используемых в организационно-экономическом управлении, является расширение сферы применения баз знаний и соответствующих им систем искусственного интеллекта.

База знаний - важнейший элемент экспертной системы, создаваемой на рабочем месте специалиста управления. Она выступает в роли накопителя знаний в конкретной области профессиональной деятельности и помощника при проведении анализа экономической ситуации в процессе выработки и принятия управленческого решения.

Информационные технологии в сфере организационно - экономического управления в настоящее время развиваются по следующим основным направлениям:

- активизация роли специалистов управления (непрофессионалов в области вычислительной техники) в подготовке и решении задач экономического управления;
- совершенствование систем интеллектуального интерфейса конечных пользователей различных уровней;
- объединение информационно-вычислительных ресурсов с помощью вычислительных сетей различных уровней (от ЛВС, объединяющих пользователей в рамках одного подразделения организации до глобальных);
- разработка комплексных мер обеспечения защиты информации (технических, организационных, программных, правовых и т.п.) от несанкционированного доступа.

1.1.2 Инструментарий информационной технологии

Техническими средствами производства информации являются аппаратное, программное и математическое обеспечение процесса. Выделим отдельно из этих средств программные продукты и назовем их инструментарием, а для большей четкости можно его конкретизировать, назвав программным инструментарием информационной технологии.

Инструментарий информационной технологии - один или несколько взаимосвязанных программных продуктов для определенного типа компьютера, технология работы в котором позволяет достичь поставленную пользователем цель.

В качестве инструментария можно использовать следующие распространенные виды программных продуктов для персонального компьютера: текстовый процессор (редактор), настольные издательские системы, электронные таблицы, системы управления базами данных, электронные записные книжки, электронные календари, информационные системы функционального назначения (финансовые, бухгалтерские, для маркетинга и пр.), экспертные системы и т.д.

1.1.3 Информационная технология и информационная система

Информационная технология тесно связана с информационными системами, которые являются для нее основной средой. На первый взгляд может показаться, что определения информационной технологии и системы очень похожи между собой.

Информационная технология является процессом, состоящим из четко регламентированных правил выполнения операций, действий, этапов разной степени сложности над данными, хранящимися в компьютерах. Основная цель информационной технологии - в результате целенаправленных действий по переработке первичной информации получить необходимую для пользователя информацию.

Информационная система представляет собой человеко-компьютерную систему обработки информации. Информационная система является средой, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технические и программные средства, связи и т.д. Основная цель информационной системы - организация хранения и передачи информации.

Реализация функций информационной системы невозможна без знания ориентированной на нее информационной технологии, Информационная технология может существовать и вне сферы информационной системы

Пример. Информационная технология работы в среде текстового процессора Microsoft Word, который не является информационной системой.

Таким образом, информационная технология является более емким понятием, отражающим современное представление о процессах преобразования информации в информационном обществе. В умелом сочетании двух информационных технологий - управленческой и компьютерной - залог успешной работы информационной системы.

Обобщая все вышесказанное, введем несколько более узкие определения информационной системы и технологии, реализованные средствами компьютерной техники.

Информационная технология - совокупность четко определенных целенаправленных действий персонала по переработке информации на компьютере.

Информационная система - человеко-компьютерная система для поддержки принятия решений и производства информационных продуктов, использующая компьютерную информационную технологию.

1.2 Этапы развития информационных технологий

Существует несколько точек зрения на развитие информационных технологий с использованием компьютеров, которые определяются различными признаками деления.

Общим для всех изложенных ниже подходов является то, что с появлением персонального компьютера начался новый этап развития информационной технологии. Основной целью становится удовлетворение персональных информационных потребностей человека как в профессиональной сфере, так и в бытовой.

По признаку - вид задач и процессов обработки информации - выделяются два этапа:

1-й этап (60 - 70-е гг.) - обработка данных в вычислительных центрах в режиме коллективного пользования. Основным направлением развития информационной технологии являлась автоматизация операционных рутинных действий человека.

2-й этап (с 80-х гг.) - создание информационных технологий, направленных на решение стратегических задач.

По признаку - проблемы, стоящие на пути информатизации общества - выделяются четыре этапа:

1-й этап (до конца 60-х гг.) характеризуется проблемой обработки больших объемов данных в условиях ограниченных возможностей аппаратных средств.

2-й этап (до конца 70-х гг.) связывается с распространением ЭВМ серии IBM/360. Проблема этого этапа - отставание программного обеспечения от уровня развития аппаратных средств.

3-й этап (с начала 80-х гг.) - компьютер становится инструментом непрофессионального пользователя, а информационные системы - средством поддержки принятия его решений. Проблемы - максимальное удовлетворение потребностей пользователя и создание соответствующего интерфейса работы в компьютерной среде.

4-й этап (с начала 90-х гг.) создание современной технологии межорганизационных связей и информационных систем. Проблемы того этапа весьма многочисленны. Наиболее существенными из них являются:

- выработка соглашений и установление стандартов, протоколов компьютерной связи;
- организация доступа к стратегической информации;
- организация защиты и безопасности информации.

По признаку - преимущество, которое приносит компьютерная технология выделяются три этапа:

1-й этап (с начала 60-х гг.) характеризуется довольно эффективной обработкой информации при выполнении рутинных операций с ориентацией на централизованное коллективное использование ресурсов вычислительных центров. Основным критерием оценки эффективности создаваемых информационных систем была разница между затраченными на разработку и сэкономленными в результате внедрения средствами.

2-й этап (с середины 70-х гг.) связан с появлением персональных компьютеров. Изменился подход к созданию информационных систем - ориентация смещается в сторону индивидуального пользователя для поддержки принимаемых им решений.

3-й этап (с начала 90-х гг.) связан с понятием анализа стратегических преимуществ в бизнесе и основан на достижениях телекоммуникационной технологии распределенной обработки информации. Информационные системы имеют своей целью не просто увеличение эффективности обработки данных и помощь управленцу. Соответствующие информационные технологии должны помочь организации выстоять в конкурентной борьбе и получить преимущество.

По признаку - виды инструментария технологии - выделяются пять этапов:

1-й этап (до второй половины XIX в.) - "ручная" информационная технология, инструментарий которой составляли: перо, чернильница, книга. Коммуникации осуществлялись ручным способом путем переправки через почту писем, пакетов, депеш.

2-й этап (с конца XIX в.) - "механическая" технология, инструментарий которой составляли: пишущая машинка, телефон, оснащенная более совершенными средствами доставки почта.

3-й этап (40 - 60-е гг. XX в.) - "электрическая" технология, инструментарий которой составляли: большие ЭВМ и соответствующее программное обеспечение, электрические пишущие машинки, ксероксы, портативные диктофоны.

4-й этап (с начала 70-х гг.) - "электронная" технология, основным инструментарием которой становятся большие ЭВМ и создаваемые на их базе автоматизированные системы управления (АСУ) и информационно-поисковые системы (ИПС), оснащенные широким спектром базовых и специализированных программных комплексов.

5-й этап (с середины 80-х гг.) - "компьютерная" ("новая") технология, основным инструментарием которой является персональный компьютер с широким спектром стандартных программных продуктов разного назначения. На этом этапе происходит процесс персонализации АСУ, который проявляется в

создании систем поддержки принятия решений определенными специалистами. Подобные системы имеют встроенные элементы анализа и интеллекта для разных уровней управления, реализуются на персональном компьютере и используют телекоммуникации.

В связи с переходом на микропроцессорную базу существенным изменениям подвергаются и технические средства бытового, культурного и прочего назначений. Начинают широко использоваться в различных областях глобальные и локальные компьютерные сети.

1.3 Особенности новых информационных технологий

Информационная технология является наиболее важной составляющей процесса использования информационных ресурсов общества. К настоящему времени она прошла несколько эволюционных этапов, смена которых определялась главным образом развитием научно-технического прогресса, появлением новых технических средств переработки информации. В современном обществе основным техническим средством технологии переработки информации служит Персональный компьютер. Внедрение персонального компьютера в информационную сферу и применение телекоммуникационных средств связи определили новый этап развития информационной технологии и, как следствие, изменение ее названия за счет присоединения одного из синонимов: "новая", "компьютерная" или "современная".

Прилагательное "новая" подчеркивает новаторский, а не Эволюционный характер этой технологии. Ее внедрение является новаторским актом в том смысле, что она существенно изменяет содержание различных видов деятельности в организациях. В понятие новой информационной технологии включены также коммуникационные технологии, которые обеспечивают передачу информации разными средствами, а именно - телефон, телеграф, телекоммуникации, факс и др.

Новая информационная технология - информационная технология с "дружественным" интерфейсом работы пользователя, использующая персональные компьютеры и телекоммуникационные средства.

Прилагательное "компьютерная" подчеркивает, что основным техническим средством ее реализации является компьютер.

Три основных принципа новой (компьютерной) информационной технологии:

- интерактивный (диалоговый) режим работы с компьютером;
- интегрированность с другими программными продуктами;
- гибкость процесса изменения как данных, так и постановок задач.

Для эффективного взаимодействия конечных пользователей с вычислительной системой новые информационные технологии опираются на принципиально иную организацию интерфейса

пользователей с вычислительной системой (так называемого дружественного интерфейса), который выражается прежде всего в следующем:

- в обеспечении права пользователя на ошибку благодаря защите информационно-вычислительных ресурсов системы от непрофессиональных действий на компьютере;
- в наличии широкого набора иерархических меню, системы подсказок и обучения и т.п., облегчающих процесс взаимодействия пользователя с ПК;
- в наличии системы "отката", позволяющей при выполнении регламентированного действия, последствия которого по каким-либо причинам не удовлетворили пользователя, вернуться к предыдущему состоянию системы.

По-видимому, более точным следует считать все же термин новая, а не компьютерная информационная технология, поскольку он отражает в ее структуре не только технологии, основанные на использовании компьютеров, но и технологии, основанные на других технических средствах, особенно на средствах, обеспечивающих телекоммуникацию.

1.4 Проблемы использования информационных технологий

Для информационных технологий является вполне естественным то, что они *устаревают* и заменяются новыми.

Например: на смену технологии пакетной обработки программ на большой ЭВМ в вычислительном центре пришла технология работы на персональном компьютере на рабочем месте пользователя. Телеграф передал все свои функции телефону и т.д.

При внедрении новой информационной технологии в организации необходимо оценить риск отставания от конкурентов в результате ее неизбежного устаревания со временем, так как информационные продукты, как никакие другие виды материальных товаров, имеют чрезвычайно высокую скорость сменяемости новыми видами, версиями. Периоды сменяемости колеблются от нескольких месяцев до одного года.

Если в процессе внедрения новой информационной технологии этому фактору не уделять должного внимания, возможно, что к моменту внедрения новой информационной технологию она уже устареет и придется принимать меры к ее модернизации. Основной причиной неудач является отсутствие или слабая проработанность методологии использования информационной технологии.

Раздел 2. ВИДЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

2.1 Классификация видов информационных технологий

В настоящее время классификация ИТ проводится по следующим признакам:

- способу реализации в автоматизированных информационных системах (АИС),
- степени охвата задач управления,
- классам реализуемых технологических операций,
- типу пользовательского интерфейса,
- вариантам использования сети ЭВМ,
- обслуживаемой предметной области и др.

1) *По способу реализации ИТ* делятся на традиционные и современные ИТ. Традиционные ИТ существовали в условиях централизованной обработки данных, до периода массового использования ПЭВМ. Они были ориентированы главным образом на снижение трудоемкости пользователя (например, инженерные и научные расчеты, формирование регулярной отчетности на предприятиях и др.). Новые (современные) ИТ связаны в первую очередь с информационным обеспечением процесса управления в режиме реального времени.

2) *По степени охвата информационными технологиями задач управления* выделяют: электронную обработку данных, автоматизацию функций управления, поддержку принятия решений, электронный офис, экспертную поддержку.

В первом случае электронная обработка данных выполняется с использованием ЭВМ без пересмотра методологии и организации процессов управления при решении локальных математических и экономических задач.

Во втором случае при автоматизации управленческой деятельности вычислительные средства используются для комплексного решения функциональных задач, формирования регулярной отчетности и работы в информационно-справочном режиме для подготовки управленческих решений. К этой же группе относятся ИТ поддержки принятия решений, которые предусматривают широкое использование экономико-математических методов и моделей, пакеты прикладных программ (ППП) для аналитической работы и формирования прогнозов, составления бизнес-планов, обоснованных оценок и выводов по процессам и явлениям производственно-хозяйственной деятельности.

К названной группе относятся и широко внедряемые в настоящее время ИТ, получившие название электронного офиса и экспертной поддержки принятия решений. Эти два варианта ИТ ориентированы на использование достижений в области новейших подходов к автоматизации работы специалистов и руководителей, создание для них наиболее благоприятных условий выполнения профессиональных функций, качественного и современного информационного обслуживания за счет автоматизированного

набора управленческих процедур, реализуемых в условиях конкретного рабочего места и офиса в целом.

Электронный офис предусматривает наличие интегрированных ППП, которые обеспечивают комплексную реализацию задач предметной области. В настоящее время все большее распространение приобретают электронные офисы, сотрудники и оборудование которых могут находиться в разных помещениях. Необходимость работы с документами, материалами и базами данных (БД) конкретного предприятия или учреждения в гостинице, транспорте, дома привела к появлению электронных офисов, включенных в соответствующие сети ЭВМ

ИТ экспертной поддержки принятия решений, составляют основу автоматизации труда специалистов-аналитиков. Эти работники кроме аналитических методов и моделей для исследования складывающихся ситуаций вынуждены использовать накопленный опыт в оценке ситуаций, т.е. сведения, составляющие базу знаний в конкретной предметной области.

3) *По классу реализуемых технологических операций ИТ подразделяются:* на работу с текстовым и табличным процессорами, графическими объектами, системы управления БД, гипертекстовые и мультимедийные системы.

Технология формирования видеоизображения получила название компьютерной графики.

Компьютерная графика - это создание, хранение и обработка моделей объектов и их изображений с помощью ЭВМ. Эта технология проникла в область моделирования различных конструкций (машиностроение, авиационная техника, автомобилестроение, строительная техника и др.), экономического анализа, проникает в рекламную деятельность, делает занимательным досуг. Формируемые и обрабатываемые с помощью цифрового процессора изображения могут быть демонстрационными и анимационными. К демонстрационным изображениям относят, как правило, коммерческую (деловую) и иллюстрационную графику. Ко второй группе - анимационной графике - принадлежит инженерная и научная графика, а также графика, связанная с рекламой, искусством, играми, когда на экран выводятся не только одиночные изображения, но и последовательность кадров в виде фильма (интерактивный вариант). Интерактивная графика является одним из наиболее прогрессивных направлений среди современных ИТ. Это направление переживает бурное развитие в области появления новых графических станций и в области специализированных программных средств, позволяющих создавать реалистические объемные движущиеся изображения, сравнимые по качеству с кадрами видеофильма.

В классическом понимании система управления БД (СУБД) представляет собой набор программ, позволяющих создавать и поддерживать БД в актуальном состоянии.

Обычно любой текст представляется как одна длинная строка символов, которая читается в одном направлении.

Гипертекстовая технология - организация текста в виде иерархической структур. Материал текста делится на фрагменты. Каждый видимый на экране ЭВМ фрагмент, дополненный многочисленными связями с другими фрагментами, позволяет уточнить информацию об изучаемом объекте и двигаться в одном или нескольких направлениях по выбранной связи.

Мультимедиа-технология - программно-техническая организация обмена с компьютером текстовой, графической, аудио и видеоинформацией.

4) **По типу пользовательского интерфейса** можно рассматривать ИТ с точки зрения возможностей доступа пользователя к информационным и вычислительным ресурсам. Так, пакетная ИТ исключает возможность пользователя влиять на обработку информации, пока она проводится в автоматическом режиме. В отличие от пакетной диалоговая ИТ предоставляет пользователю неограниченную возможность взаимодействовать с хранящимися в системе информационными ресурсами в реальном масштабе времени, получая при этом всю необходимую информацию для решения функциональных задач и принятия решений.

Интерфейс сетевой ИТ предоставляет пользователю средства доступа к территориально распределенным информационным и вычислительным ресурсам благодаря развитым средствам связи.

В настоящее время наблюдается тенденция к объединению различных типов ИТ в единый компьютерно – технологический комплекс, который носит название интегрированного. Особое место в нем принадлежит средствам коммуникации, обеспечивающим не только чрезвычайно широкие технологические возможности автоматизации управленческой деятельности, но и являющимся основой создания самых разнообразных сетевых вариантов ИТ: локальных, многоуровневых, распределенных и глобальных информационно-вычислительных сетей.

5) **По обслуживаемым предметным областям ИТ** подразделяются разнообразно. Например, только в экономике ими являются, бухгалтерский учет, банковская, налоговая и страховая деятельность и др.

Рассмотрим типичные применения информационных технологий, применяемых в управленческой системе предприятия:

Бухгалтерский учет - классическая область применения информационных технологий и наиболее часто реализуемая на сегодняшний день задача. Во-первых, ошибка бухгалтера может стоить очень дорого, поэтому очевидна выгода использования возможностей автоматизации бухгалтерии. Во-вторых, задача бухгалтерского учета довольно легко формализуется, так что разработка систем автоматизации бухгалтерского учета не представляет технически сложной проблемы.

Управление финансовыми потоками. Внедрение информационных технологий в управление финансовыми потоками также обусловлено критичностью этой области управления предприятия к ошибкам.

Неправильно построив систему расчетов с поставщиками и потребителями, можно спровоцировать кризис наличности даже при налаженной

сети закупки, сбыта и хорошем маркетинге. И наоборот, точно просчитанные и жестко контролируемые условия финансовых расчетов могут существенно увеличить оборотные средства фирмы.

Управление складом, ассортиментом, закупками. Можно автоматизировать процесс анализа движения товара.

Управление производственным процессом представляет собой очень трудоемкую задачу. Основными механизмами здесь являются планирование и оптимальное управление производственным процессом.

Автоматизированное решение подобной задачи дает возможность грамотно планировать, учитывать затраты, проводить техническую подготовку производства, оперативно управлять процессом выпуска продукции в соответствии с производственной программой и технологией.

Очевидно, что чем крупнее производство, тем большее число бизнес-процессов участвует в создании прибыли, а значит, использование информационных систем жизненно необходимо.

Управление маркетингом подразумевает сбор и анализ данных о фирмах-конкурентах, их продукции и ценовой политике, а также моделирование параметров внешнего окружения для определения оптимального уровня цен, прогнозирования прибыли и планирования рекламных кампаний. Решение большинства этих задач могут быть формализованы и представлены в виде информационной системы, позволяющей существенно повысить эффективность управления маркетингом.

Документооборот является очень важным процессом деятельности любого предприятия. Хорошо отлаженная система учетного документооборота отражает реально происходящую на предприятии текущую производственную деятельность и дает управленцам возможность воздействовать на нее. Поэтому автоматизация документооборота позволяет повысить эффективность управления.

Оперативное управление предприятием. Информационная технология, решающая задачи оперативного управления предприятием строится на основе базы данных, в которой фиксируется вся возможная информация о предприятии. Информационная система оперативного управления включает в себя массу программных решений автоматизации бизнес-процессов, имеющих место на конкретном предприятии.

Предоставление информации о фирме. Активное развитие сети Интернет привело к необходимости создания корпоративных серверов для предоставления различного рода информации о предприятии. Практически каждое уважающее себя предприятие сейчас имеет свой web-сервер. Web-сервер предприятия решает ряд задач, из которых можно выделить две основные:

- создание имиджа предприятия;
- максимальная разгрузка справочной службы компании путем предоставления потенциальным и уже существующим абонентам возможности получения необходимой информации о фирме,
- предлагаемых товарах, услугах и ценах.

Кроме того, использование web-технологий открывает широкие перспективы для электронной коммерции и обслуживания покупателей через Интернет.

2.2 Информационная технология обработки данных

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки.

Эта технология применяется на уровне исполнительской деятельности персонала невысокой квалификации в целях автоматизации некоторых рутинных постоянно повторяющихся операций управленческого труда. Поэтому внедрение информационных технологий и систем на этом уровне существенно повысит производительность труда персонала, освободит его от рутинных операций, возможно, даже приведет к необходимости сокращения численности работников.

На уровне операционной деятельности решаются следующие задачи:

- обработка данных об операциях, производимых фирмой;
- создание периодических контрольных отчетов о состоянии дел в фирме;
- получение ответов на всевозможные текущие запросы и оформление их в виде бумажных документов или отчетов.

Примеры рутинных операций: операция проверки на соответствие нормативу уровня запасов указанных товаров на складе. При уменьшении уровня запаса выдается заказ поставщику с указанием необходимого количества товара и сроков, другой пример, операция продажи товаров фирмой, в результате которой формируется выходной документ для покупателя в виде чека или квитанции.

Пример контрольного отчета: ежедневный отчет о поступлениях и выдачах наличных средств банком, формируемый в целях контроля баланса наличных средств.

Пример запроса: запрос к базе данных по кадрам, который позволит получить данные о требованиях, предъявляемых к кандидатам на занятие определенной должности.

Основные компоненты

Представим основные компоненты информационной технологии обработки данных (рис. 2.2.) и приведем их характеристики.

Примеры информационных систем оперативного уровня: бухгалтерская; банковских депозитов; обработки заказов; регистрации авиабилетов; выплаты зарплаты и т.д.

Наибольшее число пакетов прикладных программ создано для бухгалтерского учета. Среди них можно отметить "Турбо-Бухгалтер", "Инфо-Бухгалтер", "Парус", "ABACUS", и др.

Типичной информационной системой оперативного уровня является популярная программа "1С: Бухгалтерия" фирмы "1С" для Windows. Эта программа предоставляет широкие возможности манипулирования бухгалтерскими данными.

Программа "1С:Бухгалтерия" входит в комплекс программ "1 С: Предприятие", который включает также разделы "1С:Торговля+Склад" и "1С:Зарплата+Кадры".

Справочное и информационное обеспечение экономической деятельности представлено следующими пакетами программ: "ГАРАНТ" (налоги, бухучет, аудит, предпринимательство, банковское дело, валютное регулирование, таможенный контроль), "КОНСУЛЬТАНТ+" (налоги, бухучет, аудит, предпринимательство, банковское дело, валютное регулирование, таможенный контроль).

2.3 Информационная технология управления

Цель информационной технологии управления - удовлетворение информационных потребностей всех без исключения сотрудников фирмы, имеющих дело с принятием решений. Она может быть полезна на любом уровне управления.

Для принятия решений на уровне управленческого контроля информация должна быть представлена в агрегированном виде так, чтобы просматривались тенденции изменения данных, причины возникших отклонений и возможные решения. На этом этапе решаются следующие задачи обработки данных:

- оценка планируемого состояния объекта управления;
- оценка отклонений от планируемого состояния;
- выявление причин отклонений;
- анализ возможных решений и действий.

Эта технология ориентирована на работу в среде информационной системы управления и используется при худшей структурированности решаемых задач, если их сравнивать с задачами, решаемыми с помощью информационной технологии обработки данных.

ИС управления идеально подходят для удовлетворения сходных информационных потребностей работников различных функциональных подсистем (подразделений) или уровней управления фирмой. Поставляемая ими информация содержит сведения о прошлом, настоящем и

вероятном будущем фирмы. Эта информация имеет вид регулярных или специальных управленческих отчетов.

Регулярные отчеты создаются в соответствии с установленным графиком, определяющим время их создания, например месячный анализ продаж компании.

Специальные отчеты создаются по запросам управленцев или когда в компании произошло что-то незапланированное.

И те, и другие виды отчетов могут иметь форму суммирующих, сравнительных и чрезвычайных отчетов.

В суммирующих отчетах данные объединены в отдельные группы, отсортированы и представлены в виде промежуточных и окончательных итогов по отдельным полям.

Сравнительные отчеты содержат данные, полученные из различных источников или классифицированные по различным признакам и используемые для целей сравнения.

Чрезвычайные отчеты содержат данные исключительного (чрезвычайного) характера.

Использование отчетов для поддержки управления оказывается особенно эффективным при реализации так называемого управления по отклонениям.

Управление по отклонениям предполагает, что главным содержанием получаемых менеджером данных должны являться отклонения состояния хозяйственной деятельности фирмы от некоторых установленных стандартов (например, от ее запланированного состояния).

Основные компоненты

Основные компоненты информационной технологии управления показаны на рис. 2.3.

Входная информация поступает из систем операционного уровня. Выходная информация формируется в виде управленческих отчетов в удобном для принятия решения виде.

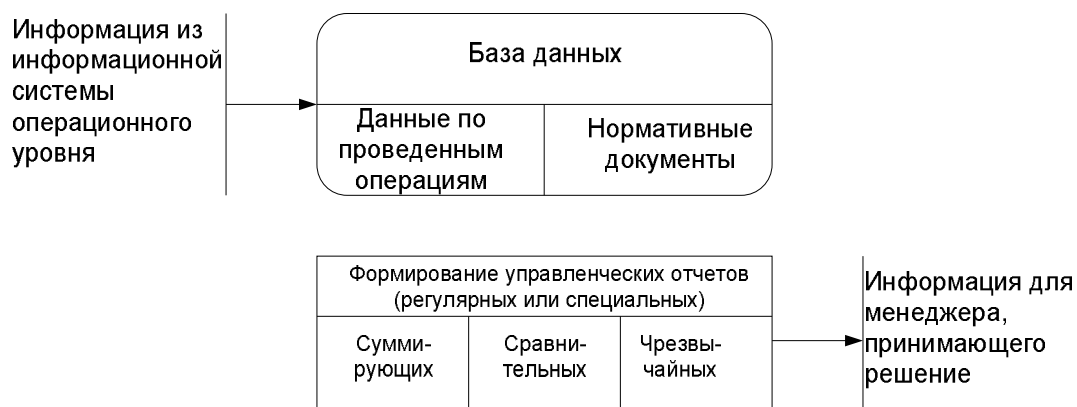


Рис.2.3 - Основные компоненты информационной технологии управления

Содержимое базы данных при помощи соответствующего программного обеспечения преобразуется в периодические и специальные отчеты, поступающие к специалистам, участвующим в принятии решений в организации. База данных, используемая для получения указанной информации, должна состоять из двух элементов:

- 1) данных, накапливаемых на основе оценки операций, проводимых фирмой;
- 2) планов, стандартов, бюджетов и других нормативных документов, определяющих планируемое состояние объекта управления (подразделения фирмы).

Пример.

Экономическая и финансовая деятельность поддерживается следующими пакетами программ:

Программный продукт Audit Expert, разработанный фирмой ПроИнвест-Консалтинг, является эффективным инструментом комплексного анализа финансового состояния и результатов деятельности предприятия. Приведение финансовой отчетности к международному стандарту позволяет Audit Expert преобразовать данные финансовой отчетности предприятий за разные годы в аналитические таблицы, соответствующие требованиям Международных стандартов бухгалтерского учета. Такой подход делает результаты анализа понятными как для российских, так и для иностранных экспертов. Отчеты формируются как на русском, так и на английском языке.

"Экономический анализ и прогноз деятельности фирмы, организации" (фирма ИНЕК), реализующий функции: экономический анализ деятельности фирмы, предприятия; бизнес-план; технико-экономическое обоснование возврата кредитов; анализ и отбор вариантов деятельности; прогноз баланса, потоков денежных средств и готовой продукции;

"Финансовый анализ предприятия" (фирма Инфософт), реализующий функции: общая оценка финансового состояния; анализ финансовой устойчивости; анализ ликвидности баланса; анализ финансовых коэффициентов (ликвидность, маневренность, покрытие, соотношение заемных и собственных средств); анализ коэффициентов деловой активности; расчет и анализ коэффициентов оборачиваемости; оценка рентабельности производства.

В области создания финансово-кредитных систем работают фирмы "Диа-софт", "Инверсия", R-Style, Программбанк, "Асофт" и др.

2.4 Автоматизация офисной деятельности

Исторически автоматизация началась на производстве и затем распространилась на офис, имея вначале целью лишь автоматизацию рутинной секретарской работы. По мере развития средств коммуникаций автоматизация офисных технологий заинтересовала специалистов и управленцев, которые увидели в ней возможность повысить производительность своего труда.

Автоматизация офиса призвана не заменить существующую традиционную систему коммуникации персонала (с ее совещаниями, телефонными звонками и приказами), а лишь дополнить ее. Совместное использование этих систем обеспечивает рациональную автоматизацию управленческого труда и наилучшее обеспечение управленцев информацией.

Автоматизированный офис привлекателен для менеджеров всех уровней управления в фирме не только потому, что поддерживает внутрифирменную связь персонала, но также потому, что предоставляет им новые средства коммуникации с внешним окружением.

Информационная технология автоматизированного офиса - организация и поддержка коммуникационных процессов как внутри организации, так и с внешней средой на базе компьютерных сетей других современных средств передачи и работы с информацией

Офисные автоматизированные технологии используются управленцами, специалистами, секретарями и конторскими служащими, особенно они привлекательны для группового решения проблем. Они позволяют повысить производительность труда секретарей и конторских работников и дают им возможность справляться с возрастающим объемом работ. Однако это преимущество является второстепенным по сравнению с возможностью использования автоматизации офиса в качестве инструмента для решения проблем. Улучшение и ускорение принимаемых менеджерами решений в результате их более совершенной коммуникации способно обеспечить экономический рост фирмы.

В настоящее время известно несколько десятков программных продуктов для компьютеров и некомпьютерных технических средств, обеспечивающих технологию автоматизации офиса: текстовый процессор, табличный процессор, электронная почта, электронный календарь, аудиопочта, компьютерные и телеконференции, видеотекст, хранение изображений, а также специализированные программы управленческой деятельности: ведения документов, контроля за исполнением приказов и т.д.

Автоматизацию офиса дополняют некомпьютерные средства: аудио и видеоконференции, факсимильная связь, ксерокс и другие средства оргтехники.

Основные компоненты

База данных является обязательным компонентом любой информационной технологии. В автоматизированном офисе база данных

концентрирует в себе данные о производственной системе фирмы так же, как в технологии обработки данных на операционном уровне. Информация в базу данных может также поступать из внешнего окружения фирмы. Специалисты должны владеть основными технологическими операциями по работе в среде баз данных.

Пример. В базе данных собираются сведения о ежедневных продажах, передаваемые торговыми агентами фирмы на главный компьютер, или сведения о еженедельных поставках сырья. Могут ежедневно по электронной почте поступать с биржи сведения о курсе валют или котировках ценных бумаг, в том числе и акций этой фирмы, которые ежедневно корректируются в соответствующем массиве базы данных.

Информация из базы данных поступает на вход компьютерных приложений (программ), таких, как текстовый процессор, табличный процессор, электронная почта, компьютерные конференции и пр. Любое компьютерное приложение автоматизированного офиса обеспечивает работникам связь друг с другом и с другими фирмами.

Текстовый процессор - это вид прикладного программного обеспечения, предназначенный для создания и обработки текстовых документов. Таким образом, в распоряжении менеджера имеется эффективный вид письменной коммуникации. Регулярное получение подготовленных с помощью текстового процессора писем и докладов дает возможность менеджеру постоянно оценивать ситуацию на фирме.

Электронная почта (E-mail), основываясь на сетевом использовании компьютеров, дает возможность пользователю получать, хранить и отправлять сообщения своим партнерам по сети. Здесь имеет место только однонаправленная связь. Это ограничение, по мнению многих исследователей, не является слишком важным, поскольку в пятидесяти случаях из ста служебные переговоры по телефону имеют целью лишь получение информации. Для обеспечения двухсторонней связи придется многократно посылать и принимать сообщения по электронной почте или воспользоваться другим способом коммуникации.

Электронная почта может предоставлять пользователю различные возможности в зависимости от используемого программного обеспечения. Чтобы посылаемое сообщение стало доступно всем пользователям электронной почты, его следует поместить на компьютерную доску объявлений, при желании можно указать, что это частная корреспонденция. Вы также можете послать отправление с уведомлением о его получении адресатом.

Аудиопочта - это почта для передачи сообщений голосом. Она напоминает электронную почту, за исключением того, что вместо набора сообщения на клавиатуре компьютера вы передаете его через телефон. Также по телефону вы получаете присланные сообщения. Система включает в себя специальное устройство для преобразования аудиосигналов в цифровой код и обратно, а также компьютер для хранения аудиосообщений в цифровой форме. Аудиопочта также реализуется в сети. Система будет периодически обзванивать всех указанных сотрудников для передачи им сообщения.

Табличный процессор так же, как и текстовый процессор, является базовой составляющей информационной культуры любого сотрудника и автоматизированной офисной технологии. Без знания основ технологии работы в нем невозможно полноценно использовать персональный компьютер в своей деятельности. Функции современных программных сред табличных процессоров позволяют выполнять многочисленные операции над данными, представленными в табличной форме. Объединяя эти операции по общим признакам, можно выделить наиболее многочисленные и применяемые группы технологических операций:

- ввод данных как с клавиатуры, так и из баз данных;
- обработка данных (сортировка, автоматическое формирование итогов, копирование и перенос данных, различные группы операций по вычислениям, агрегирование данных и т.д.);
- вывод информации в печатном виде, в виде импортируемых файлов в другие системы, непосредственно в базу данных;
- качественное оформление табличных форм представления данных;
- многоплановое и качественное оформление данных в виде диаграмм и графиков;
- проведение инженерных, финансовых, статистических расчетов;
- проведение математического моделирования и ряд других вспомогательных операций.

Любая современная среда табличного процессора имеет средства пересылки данных по сети.

Электронный календарь предоставляет еще одну возможность использовать сетевой вариант компьютера для хранения и манипулирования рабочим расписанием управленцев и других работников организации. Менеджер (или его секретарь) устанавливает дату и время встречи или другого мероприятия, просматривает получившееся расписание, вносит изменения при помощи клавиатуры. Техническое и программное обеспечение электронного календаря полностью соответствует аналогичным компонентам электронной почты. Более того, программное обеспечение календаря часто является составной частью программного обеспечения электронной почты.

Система дополнительно дает возможность получить доступ также и к календарям других менеджеров. Она может автоматически согласовать время встречи с их собственными расписаниями.

Использование электронного календаря оказывается особенно эффективным для менеджеров высших уровней управления, рабочие дни которых расписаны надолго вперед.

Компьютерные конференции используют компьютерные сети для обмена информацией между участниками группы, решающей определенную проблему. Естественно, круг лиц, имеющих доступ к этой технологии, ограничен. Количество участников компьютерной конференции может быть во много раз больше, чем аудио- и видеоконференций.

Телеконференция включает в себя три типа конференций: аудио, видео и компьютерную. Видеотекст основан на использовании компьютера для получения отображения текстовых и графических данных на экране монитора.

Обмен каталогами и ценниками (прайс-листами) своей продукции между компаниями в форме видеотекста приобретает сейчас все большую популярность. Что же касается компаний, специализирующихся на продаже видеотекста, то их услуги начинают конкурировать с такой печатной продукцией, как газеты и журналы. Так, во многих странах сейчас можно заказать газету или журнал в форме видеотекста, не говоря уже о текущих сводках биржевой информации.

Хранение изображений. В любой фирме необходимо длительное время хранить большое количество документов. Их число может быть так велико, что хранение даже в форме файлов вызывает серьезные проблемы. Поэтому возникла идея хранить не сам документ, а его образ (изображение), причем хранить в цифровой форме. Хранение изображений (imaging) является перспективной офисной технологией и основывается на использовании специального устройства

оптического распознавателя образов, позволяющего преобразовывать изображение документа или фильма в цифровой вид для дальнейшего хранения во внешней памяти компьютера. Сохраненное в цифровом формате изображение может быть в любой момент выведено в его реальном виде на экран или принтер. Для хранения изображений используются оптические диски

Идея хранения изображений не нова и реализовывалась раньше на основе микрофильмов. Созданию данной технологии способствовало появление нового технического решения - оптического диска в комбинации с цифровой записью изображения.

Аудиоконференции используют аудиосвязь для поддержания коммуникаций между территориально удаленными работниками или подразделениями фирмы. Наиболее простым техническим средством реализации аудиоконференций является телефонная связь, оснащенная дополнительными устройствами, дающими возможность участия в разговоре более чем двум участникам. Создание аудиоконференций не требует наличия компьютера, а лишь предполагает использование двухсторонней аудиосвязи между ее участниками. Использование аудиоконференций облегчает принятие решений, оно дешево и удобно.

Видеоконференции предназначены для тех же целей, что и аудиоконференций, но с применением видеоаппаратуры. Их проведение также не требует компьютера. В процессе видеоконференции ее участники, удаленные друг от друга на значительное расстояние, могут видеть на телевизионном экране себя и других участников. Одновременно с телевизионным изображением передается звуковое сопровождение.

Хотя видеоконференции позволяют сократить транспортные и командировочные расходы, большинство фирм применяет их не только по этой причине. Эти фирмы видят в них возможность привлечь к решению проблем

максимальное количество менеджеров и других работников, территориально удаленных от главного офиса.

Факсимильная связь основана на использовании факс-аппарата, способного читать документ на одном конце коммуникационного канала и воспроизводить его изображение на другом. Факсимильная связь вносит свой вклад в принятие решений за счет быстрой и легкой рассылки документов участникам группы, решающей определенную проблему, независимо от их географического положения.

Наиболее популярным набором программ для офисной автоматизации является Microsoft Office. Продукты Microsoft Office тесно интегрированы между собой, они имеют более 50% общего программного кода. Это является основой однотипной работы со всеми приложениями.

2.5 Информационная технология поддержки принятия решений

Системы поддержки принятия решений и соответствующая им информационная технология появились усилиями в основном американских ученых в конце 70-х - начале 80-х гг., чему способствовали широкое распространение персональных компьютеров, стандартных пакетов прикладных программ, а также успехи в создании систем искусственного интеллекта.

Главной особенностью информационной технологии поддержки принятия решений является качественно новый метод организации взаимодействия человека и компьютера. Выработка решения, что является основной целью этой технологии, происходит в результате итерационного процесса, в котором участвуют:

- система поддержки принятия решений (СППР) в роли вычислительного звена и объекта управления;
- лица, принимающего решение, оценивающего полученный результат вычислений на компьютере.

Окончание итерационного процесса происходит по воле человека. В этом случае можно говорить о способности информационной системы совместно с пользователем создавать новую информацию для принятия решений.

Дополнительно к этой особенности информационной технологии поддержки принятия решений можно указать еще ряд ее отличительных характеристик:

- ориентация на решение плохо структурированных (формализованных) задач;
- сочетание традиционных методов доступа и обработки компьютерных данных с возможностями математических моделей и методами решения задач на их основе;
- направленность на непрофессионального пользователя компьютера;
- высокая адаптивность, обеспечивающая возможность приспосабливаться к особенностям имеющегося технического и программного обеспечения, а также требованиям пользователя.

Информационная технология поддержки принятия решений может использоваться на любом уровне управления. Кроме того, решения, принимаемые на различных уровнях управления, часто должны координироваться. Поэтому важной функцией и систем, и технологий является координация лиц, принимающих решения, как на разных уровнях управления, так и на одном уровне.

Основные компоненты

Рассмотрим структуру системы поддержки принятия решений (рис. 2.4), а также функции составляющих ее блоков, которые определяют основные технологические операции.

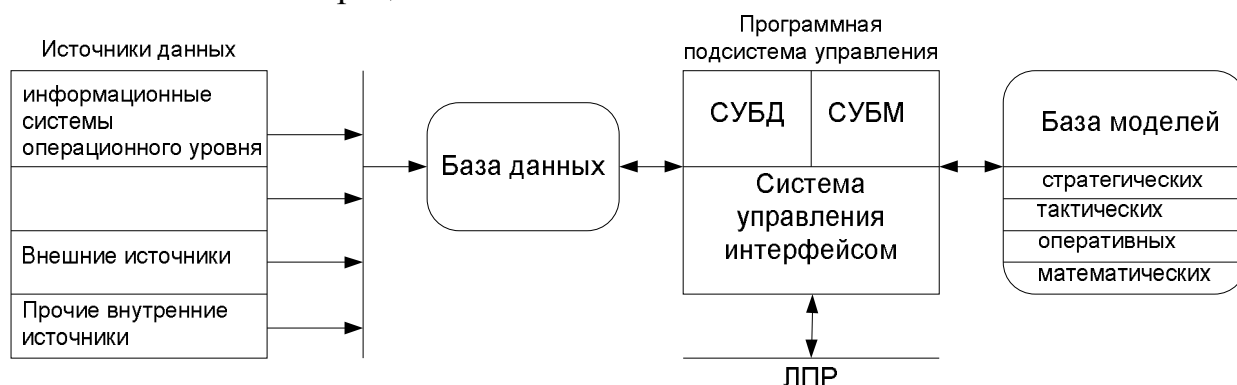


Рис. 2.4 - Основные компоненты информационной технологии поддержки принятия решений

В состав системы поддержки принятия решений входят три главных компонента: база данных, база моделей и программная подсистема, которая состоит из системы управления базой данных (СУБД), системы управления базой моделей (СУБМ) и системы управления интерфейсом между пользователем и компьютером.

База данных играет в информационной технологии поддержки принятия решений важную роль. Данные могут использоваться непосредственно пользователем для расчетов при помощи математических моделей. Рассмотрим источники данных и их особенности.

1) Часть данных поступает от информационной системы операционного уровня. Чтобы использовать их эффективно, эти данные должны быть предварительно обработаны.

2) Помимо данных об операциях фирмы для функционирования системы поддержки принятия решений требуются и другие внутренние данные, например данные о движении персонала, инженерные данные и т.п., которые должны быть своевременно собраны, введены и поддержаны.

3) Важное значение, особенно для поддержки принятия решений на верхних уровнях управления, имеют данные из внешних источников. В числе необходимых внешних данных следует указать данные о конкурентах, национальной и мировой экономике. В отличие от внутренних данных внешние данные обычно приобретаются у специализирующихся на их сборе организаций.

4) В настоящее время широко исследуется вопрос о включении в базу данных еще одного источника данных - документов, включающих в себя записи, письма, контракты, приказы и т.п. Если содержание этих документов будет записано в памяти и затем обработано по некоторым ключевым характеристикам (поставщикам, потребителям, датам, видам услуг и др.), то система получит новый мощный источник информации.

База моделей. Целью создания моделей являются описание и оптимизация некоторого объекта или процесса. Использование моделей обеспечивает проведение анализа в системах поддержки принятия решений. Модели, базируясь на математической интерпретации проблемы, при помощи определенных алгоритмов способствуют нахождению информации, полезной для принятия правильных решений.

Пример. Модель линейного программирования дает возможность определить наиболее выгодную производственную программу выпуска нескольких видов продукции при заданных ограничениях на ресурсы.

Использование моделей в составе информационных систем началось с применения статистических методов и методов финансового анализа, которые реализовывались командами обычных алгоритмических языков. Позже были созданы специальные языки, позволяющие моделировать ситуации типа "что будет, если?" или "как сделать, чтобы?". Такие языки, созданные специально для построения моделей, дают возможность построения моделей определенного типа, обеспечивающих нахождение решения при гибком изменении переменных.

Существует множество типов моделей и способов их классификации, например, по цели использования, области возможных приложений, способу оценки переменных и т. п.

По цели использования модели подразделяются на оптимизационные, связанные с нахождением точек минимума или максимума некоторых показателей (например, управляющие часто хотят знать, какие их действия ведут к максимизации прибыли или минимизации затрат), и описательные, описывающие поведение некоторой системы и не предназначенные для целей управления (оптимизации).

По способу оценки модели классифицируются на детерминированные, использующие оценку переменных одним числом при конкретных значениях исходных данных, и стохастические, оценивающие переменные несколькими параметрами, так как исходные данные заданы вероятностными характеристиками.

Детерминированные модели более популярны, потому что они менее дорогие, их легче строить и использовать. К тому же часто с их помощью получается вполне достаточная информация для принятия решения.

По области возможных приложений модели разбиваются на специализированные, предназначенные для использования только одной системой, и универсальные - для использования несколькими системами.

Специализированные модели более дорогие, они обычно применяются для описания уникальных систем и обладают большей точностью.

В системах поддержки принятия решения база моделей состоит из стратегических, тактических и оперативных моделей, а также математических моделей в виде совокупности модельных блоков, модулей и процедур, используемых как элементы для их построения (см. рис.6).

Стратегические модели используются на высших уровнях управления для установления целей организации, объемов ресурсов, необходимых для их достижения, а также политики приобретения и использования этих ресурсов. Они могут быть также полезны при выборе вариантов размещения предприятий, прогнозировании политики конкурентов и т.п. Для стратегических моделей характерны значительная широта охвата, множество переменных, представление данных в сжатой агрегированной форме. Часто эти данные базируются на внешних источниках и могут иметь субъективный характер. Горизонт планирования в стратегических моделях, как правило, измеряется в годах. Эти модели обычно детерминированные, описательные, специализированные для использования на одной определенной фирме.

Тактические модели применяются управляющими (менеджерами) среднего уровня для распределения и контроля использования имеющихся ресурсов. Среди возможных сфер их использования следует указать: финансовое планирование, планирование требований к работникам, планирование увеличения продаж, построение схем компоновки предприятий. Эти модели применимы обычно лишь к отдельным частям фирмы (например, к системе производства и сбыта) и могут также включать в себя агрегированные показатели. Временной горизонт, охватываемый тактическими моделями, - от одного месяца до двух лет. Здесь также могут потребоваться данные из внешних источников, но основное внимание при реализации данных моделей должно быть уделено внутренним данным фирмы. Обычно тактические модели реализуются как детерминированные, оптимизационные и универсальные.

Оперативные модели используются на низших уровнях управления для поддержки принятия оперативных решений с горизонтом, измеряемым днями и неделями. Возможные применения этих моделей включают в себя ведение дебиторских счетов и кредитных расчетов, календарное производственное планирование, управление запасами и т.д. Оперативные модели обычно используют для расчетов внутрифирменные данные. Они, как правило, детерминированные, оптимизационные и универсальные (т.е. могут быть использованы в различных организациях).

Математические модели состоят из совокупности модельных блоков, модулей и процедур, реализующих математические методы. Сюда могут входить процедуры линейного программирования, статистического анализа временных рядов, регрессионного анализа и т.п. - от простейших процедур до сложных ППП.

Пример.

Программный продукт Forecast Expert, также разработанный фирмой Про-Инвест-Консалтинг, представляет собой универсальную систему прикладного

прогнозирования. Forecast Expert предназначен для построения прогноза временного ряда. В качестве прогнозируемых могут выступать параметры как сфер производства и обращения - цены мирового рынка, спрос на изделия, объемы закупок комплектующих и производственных запасов при увеличении объема производства, цены комплектующих, параметры технологических процессов, так и финансового рынка - цены покупки и продажи акций, деловая активность участников рынка, объем предложений свободных средств инвесторами и многое другое.

Применение Forecast Expert позволяет проанализировать имеющиеся данные и построить прогноз с указанием границ доверительного интервала (при заданной вероятности прогноза) на период времени. Модель определяет степень влияния сезонных факторов и учитывает их при построении прогноза.

2.6 Экспертные системы

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки экспертных систем (ЭС), основанных на использовании элементов искусственного интеллекта. Экспертные системы дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, на основе которых этими системами накоплены знания.

Под **искусственным интеллектом** (ИИ) обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека. Чаще всего здесь имеются в виду способности, связанные с человеческим мышлением. Работы в области искусственного интеллекта не ограничиваются экспертными системами. Они также включают в себя создание роботов, систем, моделирующих нервную систему человека, его слух, зрение, обоняние, способность к обучению.

Решение специальных задач требует специальных знаний. Главная идея использования технологии экспертных систем заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость. Являясь одним из основных приложений искусственного интеллекта, экспертные системы представляют собой компьютерные программы, трансформирующие опыт экспертов в какой-либо области знаний в форму эвристических правил. На практике ЭС используются прежде всего как системы-советчики в тех ситуациях, где специалист сомневается в выборе правильного решения. Экспертные знания, хранящиеся в памяти системы, более глубокие и полные, чем соответствующие знания пользователя.

ЭС находят распространение при решении задач с принятием решений в условиях неопределенности (неполноты) для распознавания образов, в прогнозировании, диагностике, планировании, управлении, конструировании и т.д.

Типичная экспертная система состоит из решателя (интерпретатора), БД (базы данных), БЗ (базы знаний), компонентов приобретения знаний, объяснительного и диалогового компонентов.

БД предназначена для хранения исходных и промежуточных данных, используемых для решения задач, фактографических данных.

Решатель, используя исходные данные из БД и знания из БЗ, обеспечивает решение задач для конкретных ситуаций.

Компонент приобретения знаний автоматизирует процесс наполнения БЗ.

Объяснительный компонент объясняет, как система получила решение задачи (или почему не получила) и какие знания она при этом использовала. Диалоговый компонент обеспечивает диалог между экспертной системой и пользователем в процессе решения задачи и приобретения знаний.

Экспертные системы создаются для решения разного рода задач профессиональной деятельности человека, и в зависимости от этого выполняют разные функции.

2.6.1 Типы экспертных систем

Можно назвать несколько *типов современных экспертных систем*.

1) Экспертные системы первого поколения. Предназначены для решения хорошо структурированных задач, требующих небольшого объема эмпирических знаний. Сюда относятся классификационные задачи и задачи выбора из имеющегося набора вариантов.

2) Оболочки ЭС. Имеют механизм ввода-вывода, но БЗ пустая. Требуется настройка на конкретную предметную область. Знания приобретаются в процессе функционирования ЭС, способной к самообучению.

3) Гибридные ЭС. Предназначены для решения различных задач с использованием БЗ. Это задачи с использованием методов системного анализа, исследования операций, математической статистики, обработки информации. Пользователь имеет доступ к объективизированным знаниям, содержащимся в БЗ и пакетах прикладных программ.

4) Сетевые ЭС. Между собой связаны несколько экспертных систем. Результаты решения одной из них являются исходными данными для другой системы. Эффективны при распределенной обработке информации.

При разработке экспертных систем должны участвовать: эксперт той предметной области, задачи которой будет решать система; инженер по знаниям - специалист по разработкам систем; программист - специалист по разработке инструментальных средств. Эксперт определяет знания, то есть описывает предметную область в виде совокупности данных и правил, обеспечивает полноту и правильность введенных в экспертную систему знаний. Данные определяют объекты, их характеристики и значения. Правила указывают на способы манипулирования данными.

Инженер по знаниям помогает эксперту: выявить и структурировать знания, необходимые для функционирования экспертной системы; осуществить выбор инструментальных средств, которые наиболее эффективны для решения задач в данной предметной области; указать способы представления знаний. Программист разрабатывает инструментальную среду, включающую все компоненты экспертной системы, производит ее сопряжение с другими существующими системами.

2.6.2 Виды знаний

1) *Понятийные знания.* Это набор понятий, которыми пользуется ЛПР, работающий в некоторой области интеллектуальной, управляющей деятельности, а также свойства и взаимосвязи этих понятий. Эта категория знаний в основном вырабатывается в сфере фундаментальных наук.

2) *Конструктивные знания* (близкие к понятийным знаниям). Это знания о структуре и взаимодействии частей различных объектов. Они в основном составляют содержание технических, прикладных наук. К примеру, если взять программирование, то понятийное знание - знание о структуре операторов, данных, языка программирования. Конструктивное знание - это знание об устройстве конкретных программ, о типичных алгоритмах.

3) *Процедурные знания.* К ним относятся методические правила решения различных задач, с которыми ЛПР уже сталкивался и их решать. В производственной сфере аналогом процедурных знаний являются технологические знания различных производственных процессов. Процедурные знания - это опыт интеллектуальной, управляющей деятельности ЛПР в определенной предметной области.

4) *Фактографические знания.* Они включают в себя количественные и качественные характеристики конкретных объектов, явлений и их элементов. Их накопление ведется в виде таблиц, справочников, файлов, БД.

2.6.3 Способы формализованного представления знаний в БЗ

Формализованное представление знаний в информационных технологиях управления в виде интеллектуальных систем является первичным. Рассмотрим распространенные способы их формализованного представления.

1) **Представление знаний продукционными правилами.** Продукционные правила представляют знания в форме ЕСЛИ - ТО. Системы, использующие представления знаний продукционными правилами, называются продукционными. Это самый наглядный и простой способ. В таких системах представления знаний имеются средства, позволяющие использовать в данных и правилах нечеткую информацию с определенной вероятностью, называемой фактором уверенности.

2) **Логика предикатов (раздел математической логики).**

Константы и переменные определяют отдельные объекты и обозначаются буквами или набором букв (U, V, W, X, Y). Последовательность из n констант или переменных (n - конечно, $n > 1$) называется функцией. Атомарным предикатом называется последовательность из n сущностей и понятий, описанных константами, переменными или функциями.

Предикат принимает одно из двух значений: истина или ложь. Предикат, в котором все переменные, константы и функции связаны между собой, называется предложением. Предложения используются для представления знаний. Логика предикатов обеспечивает высокий уровень модульности знаний (представляет их как единое целое в определенной предметной области) и позволяет выяснить, имеются ли или отсутствуют противоречия между новыми и уже существующими знаниями. Но чрезмерный уровень формализации представления знаний, трудность их прочтения снижают эффективность обработки. Кроме этого, в логике предикатов все отношения описываются предикатами, что не позволяет при компьютерной обработке полностью отразить свойства структуры данных. Для программирования используется язык логического типа ПРОЛОГ.

3) **Модель доски объявлений.** Модель представляется как совокупность отдельных проблем, каждая из которых составляет отдельное множество знаний. Все множества модели используются согласованно как единое целое и управляются через общую рабочую область памяти, называемую доской объявлений. Отдельное множество знаний называется источником знаний (ИЗ), и каждый ИЗ строится как продукционная система.

4) **Семантические сети.** Знания можно рассматривать как отношения между понятиями и сущностями, являющимися конкретными объектами реального мира. Понятия и отношения можно представить в виде семантической сети, состоящей из вершин и дуг. В вершинах располагаются понятия, а направленные связи между вершинами соответствуют различного рода отношениям между этими понятиями. Семантические сети могут быть выполнены обучаемыми и растущими, что означает возможность автоматического добавления в сеть новых узлов по мере появления в опыте ее использования новых понятий, а также увеличение весовых коэффициентов, соответствующих дугам. В процессе ее обучения между существующими узлами также могут устанавливаться дополнительные связи.

5) **Фреймовые системы.** Фреймы рассматриваются как структура описания отдельной сущности или понятия. Они могут быть в виде их совокупностей, представляемых как отдельное множество знаний, относящихся к одному объекту. Каждый фрейм состоит из множества элементов, называемых слотами, которые в свою очередь представляются определенной структурой данных. Каждый фрейм и слот имеют имя, единственное во всей фреймовой системе. В значение слота содержится конкретную информацию.

Фреймы не связаны в сеть. Управление большим числом источников знаний выполняется самим пользователем путем вызова нужных процедур (в

других способах это выполняет сама система). Для поиска нужного объекта задаются значения слотов. Если данные удовлетворяют условиям всех слотов, то объект считается найденным.

2.6.4 Области применения ЭС

ЭС в задачах интерпретации, как правило, используют информацию от датчиков для описания ситуации. В качестве примера приведем интерпретацию показаний измерительных приборов на химическом заводе для определения состояния процесса. Интерпретирующие системы имеют дело не с четкими символьными представлениями проблемной ситуации, а непосредственно с реальными данными. Они сталкиваются с затруднениями, которых нет у систем других типов, потому что им приходится обрабатывать информацию зашумленную, недостаточную, неполную, ненадежную или ошибочную. Им необходимы специальные методы регистрации характеристик непрерывных потоков данных, сигналов или изображений и методы их символьного представления.

Интерпретирующие ЭС могут обработать разнообразные виды данных. Например, системы анализа сцен и распознавания речи, используя естественную информацию, - в одном случае визуальные образы, в другом - звуковые сигналы, - анализируют их характеристики и понимают их смысл. Интерпретация в области химии использует данные дифракции рентгеновских лучей, спектрального анализа или ядерно-магнитного резонанса для вывода химической структуры веществ. Интерпретирующая система в геологии использует каротажное зондирование - измерение проводимости горных пород в буровых скважинах и вокруг них, - чтобы определить подповерхностные геологические структуры. Медицинские интерпретирующие системы используют показания следящих систем (например, значения пульса, кровяного давления), чтобы установить диагноз или тяжесть заболевания. Наконец, в военном деле интерпретирующие системы используют данные от радаров, радиосвязи и сонарных устройств, чтобы оценить ситуацию и идентифицировать цели.

ЭС в задачах прогнозирования определяют вероятные последствия заданных ситуаций. Примерами служат прогноз ущерба урожаю от некоторого вида вредных насекомых, оценивание спроса на нефть на мировом рынке в зависимости от складывающейся геополитической ситуации и прогнозирование места возникновения следующего вооруженного конфликта на основании данных разведки. Системы прогнозирования иногда используют имитационное моделирование, т.е. программы, которые отражают причинно-следственные взаимосвязи в реальном мире, чтобы сгенерировать ситуации или сценарии, которые могут возникнуть при тех или иных входных данных. Эти возможные ситуации вместе со знаниями о процессах, порождающих эти ситуации, образуют предпосылки для прогноза.

ЭС в задачах диагностики используют описания ситуаций, характеристики поведения или знания о конструкции компонент, чтобы установить вероятные причины неправильного функционирования диагностируемой системы. Примерами служат: определение причин заболевания по симптомам, наблюдаемым у пациентов; локализация неисправностей в электронных схемах и определение неисправных компонент в системе охлаждения ядерных реакторов. Диагностические системы часто являются консультантами, которые не только ставят диагноз, но также помогают в отладке. Они могут взаимодействовать с *пользователем*, чтобы оказать помощь при поиске неисправностей, а затем предложить порядок действий по их устранению. Медицина представляется вполне естественной областью для диагностирования, и действительно, в медицинской области было разработано больше диагностических систем, чем в любой другой отдельно взятой предметной области.

ЭС, применяемые в области проектирования, разрабатывают конфигурации объектов с учетом набора ограничений, присущих проблеме. Учитывая то, что проектирование столь тесно связано с планированием, многие проектирующие системы содержат механизмы разработки и уточнения планов для достижения желаемого проекта. Наиболее часто встречающиеся области применения планирующих ЭС - химия, электроника и военное дело.

ЭС, которые используются для решения задач наблюдения, сравнивают действительное поведение с ожидаемым поведением системы. Примерами могут служить слежение за показаниями измерительных приборов в ядерных реакторах с целью обнаружения аварийных ситуаций или оценку данных мониторинга больных, помещенных в блоки интенсивной терапии. Наблюдающие ЭС подыскивают наблюдаемое поведение, которое подтверждает их ожидания относительно нормального поведения или их предположения о возможных отклонениях. Наблюдающие ЭС по самой своей природе должны работать в режиме реального времени.

ЭС в задачах отладки находят рецепты для исправления неправильного поведения устройств. Примерами могут служить настройка компьютерной системы с целью преодолеть некоторый вид затруднений в ее работе; выбор типа обслуживания, необходимого для устранения неисправностей в телефонном кабеле; выбор ремонтной операции для исправления известной неисправности в насосе.

ЭС в задачах ремонта аппаратуры следуют плану, который предписывает некоторые рецепты восстановления. Примером является настройка масс-спектрометра, т.е. установка ручек регулировки прибора в положение, обеспечивающее достижение оптимальной чувствительности, совместимой с правильным отношением величин пиков и их формы. Пока что было разработано очень мало ремонтных ЭС отчасти потому, что необходимость фактического выполнения ремонтных процедур на объектах реального мира дополнительно усложняет задачу. Ремонтным системам также необходимы диагностирующие, отлаживающие и планирующие процедуры для производства ремонта.

ЭС в области обучения подвергают диагностике, "отладке" и исправлению ("ремонту") поведение обучаемого. В качестве примеров приведем обучение студентов отысканию неисправностей в электрических цепях, обучение военных моряков обращению с двигателем на корабле и обучение студентов-медиков выбору антимикробной терапии. Обучающие системы создают модель того, что обучающийся знает и как он эти знания применяет к решению проблемы. Системы диагностируют и указывают обучающемуся его ошибки, анализируя модель и строя планы исправлений указанных ошибок. Они исправляют поведение обучающихся, выполняя эти планы с помощью непосредственных указаний обучающимся.

ЭС в задачах управления адаптивно руководят поведением системы в целом. Примерами служат управление производством и распределением компьютерных систем или контроль за состоянием больных при интенсивной терапии. Управляющие ЭС должны включать наблюдающие компоненты, чтобы отслеживать поведение объекта на протяжении времени, но они могут нуждаться также и в других компонентах для выполнения любых или всех из уже рассмотренных типов задач: интерпретации, прогнозирования, диагностики, проектирования, планирования, отладки, ремонта и обучения. Типичная комбинация задач состоит из наблюдения, диагностики, отладки, планирования и прогноза.

3.1 Модели информационных процессов передачи, обработки, накопления данных

3.1.1 Обобщенная схема технологического процесса обработки информации

При производстве информационного продукта исходный информационный ресурс в соответствии с поставленной задачей подвергается в определенной последовательности различным преобразованиям. Динамика этих преобразований отображается в протекающих при этом информационных процессах. Таким образом, **информационный процесс** - это процесс преобразования информации. В результате информация может изменить и содержание, и форму представления.

Управляющие воздействия формируются на основе накопленной и функционирующей в системе управления информации, а также поступающих по каналам прямой и обратной связи сведений из внешней среды.

Таким образом, важнейшая функция любой системы управления - получение информации, выполнение процедур по ее обработке с помощью заданных алгоритмов и программ, формирование на основе полученных сведений управленческих решений, определяющих дальнейшее поведение системы.

Поскольку информация фиксируется и передается на материальных носителях, необходимы действия человека и работа технических средств по восприятию, сбору информации, ее записи, передаче, преобразованию, обработке, хранению, поиску и выдаче. Эти действия обеспечивают нормальное протекание информационного процесса и входят в технологию управления. Они реализуются технологическими процессами обработки данных с использованием электронных вычислительных машин и других технических средств.

Фазы преобразования информации в информационной технологии достаточно многочисленны. Однако если провести структуризацию технологии, обобщенная схема технологического процесса обработки информации может быть представлена схемой, показанной на рис. 3.1.

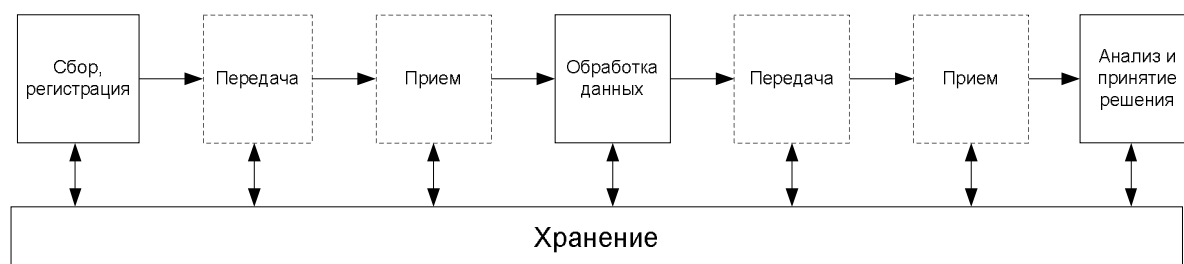


Рис.3.1 – Обобщенная схема технологического процесса обработки информации

При обработке данных формируются четыре основных информационных процесса: сбор и регистрация, обмен, обработка, накопление и хранение информации. Рассмотрим их модели.

3.1.2 Сбор и регистрация информации

Сбор и регистрация информации происходят по-разному и в различных объектах.

Процесс перевода информации в выходные данные в технологических системах управления может быть полностью автоматизирован, так как для сбора информации о состоянии производственной линии применяются разнообразные электрические датчики, которые уже по своей природе позволяют проводить преобразования физических параметров, вплоть до превращения их в данные, записываемые на машинных носителях информации, без выхода на человеческий уровень представления. Это оказывается возможным благодаря относительной простоте и однозначности физической информации, снимаемой датчиками (давление, температура, скорость и т.п.).

В организационно-экономических системах управления информация, осведомляющая человека о состоянии объекта управления семантически сложна, разнообразна и ее сбор не удастся автоматизировать. Поэтому в таких системах информационная технология на этапе превращения исходной (первичной) информации в данные в основе своей остается ручной. На рис.3.2 приведена последовательность фаз процесса преобразования информации в данные в информационной технологии организационно-экономических систем управления.

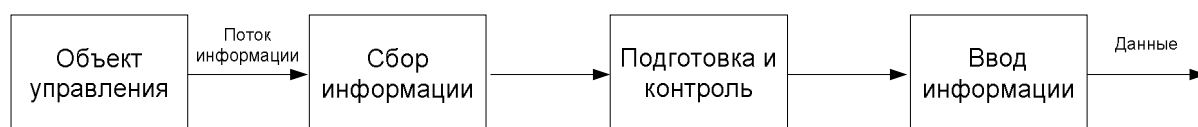


Рис.3.2 - Процесс преобразования информации в данные

Сбор информации состоит в том, что поток осведомляющей информации, поступающей от объекта управления, воспринимается человеком и переводится в документальную форму (записывается на бумажный носитель информации). Составляющими этого потока могут быть показания приборов (например, пробег автомобиля по спидометру), накладные, акты, ордера, ведомости, журналы, описи и т.д.

Для перевода потока осведомляющей информации в автоматизированный контур информационной технологии необходимо собранную информацию передать в места ее ввода в компьютер, так как часто пункты получения первичной информации от них пространственно

удалены. Передача осуществляется, как правило, традиционно, с помощью курьера, телефона.

Собранная информация для ввода должна быть предварительно подготовлена, поскольку модель предметной области, заложенная в компьютер, накладывает свои ограничения на состав и организацию вводимой информации. В современных информационных системах ввод информации осуществляется по запросам программы, отображаемым на экране дисплея, и часто дальнейший ввод приостанавливается, если оператор проигнорирован какой-либо важный запрос. Очень важными на этапах подготовки информации и ввода являются процедуры контроля.

Контроль подготовленной и вводимой информации направлен на предупреждение, выявление и устранение ошибок, которые неизбежны в первую очередь из-за так называемого "человеческого фактора". Человек устает, его внимание может ослабнуть, кто-то может его отвлечь - в результате возникают ошибки. Ошибки при сборе данных и подготовке информации могут быть и преднамеренными. Любые ошибки приводят к искажению вводимых данных, к их недостоверности, а значит, к неверным результатам обработки и в конечном итоге к ошибкам в управлении системой. При контроле собранных данных и подготовленной информации применяют совокупность приемов, как ручных, так и формализованных, направленных на обнаружение ошибок.

Вообще процедуры контроля полноты и достоверности информации и данных используются при реализации информационных процессов повсеместно и могут быть подразделены на *визуальные, логические и арифметические*.

Визуальный метод широко используется на этапе сбора и подготовки начальной информации и является ручным. При визуальном методе производится зрительный просмотр документа в целях проверки полноты, актуальности, подписей ответственных лиц, юридической законности и т.д.

Логический и арифметический, являясь автоматизированными методами, применяются на последующих этапах преобразования данных.

Логический метод контроля предполагает сопоставление фактических данных с нормативными или с данными предыдущих периодов обработки, проверку логической непротиворечивости функционально-зависимых показателей и их групп и т.д.

Арифметический метод контроля включает подсчет контрольных сумм по строкам и столбцам документов, имеющих табличную форму, контроль по формулам, признакам делимости или четности, балансовые методы, повторный ввод и т.п.

Для предотвращения случайного или намеренного искажения информации служат и организационные, и специальные мероприятия. Это четкое распределение прав и обязанностей лиц, ответственных за сбор, подготовку, передачу и ввод информации в системе информационной

технологии. Это и автоматическое протоколирование ввода, и обеспечение санкционированного доступа в контур ИТ.

В настоящее время в нашей стране, как и во всем мире, персональные компьютеры все шире применяются на рабочих местах служащих, ответственных за сбор, подготовку и предварительный контроль первичной информации. В этом случае используются автоматизированные подготовка и контроль собранной необработанной информации и, *таким образом, фазы подготовки и ввода объединяются.*

Ввод первоначальной информации при создании информационной технологии в организационно-экономической системе в конечном итоге является ручным - пользователь ЭВМ "набирает" данные (алфавитно-цифровые) на клавиатуре, визуально контролируя правильность вводимых символов по отображению на экране дисплея. Каждое нажатие клавиши - это преобразование символа, изображенного на ней, в электрический двоичный код, т.е. в машинные данные. Этап ввода - заключительный этап процесса преобразования исходной информации в машинные данные. Конечно, сейчас есть, помимо клавиатуры, и другие устройства ввода, позволяющие убыстрить и упростить этот трудоемкий и изобилующий ошибками этап, например сканеры или устройства ввода с голоса. Однако указанные устройства, особенно последние, далеки от совершенства и имеют довольно высокую стоимость.

Для решения задач ИТ, помимо ввода осведомляющей информации об объекте управления, необходимо также подготавливать и вводить информацию о структуре и содержании предметной области (т.е. модель объекта управления), а также информацию о последовательности и содержании процедур технологических преобразований для решения поставленных задач (т.е. алгоритмическую модель). Суть подготовки информации такого вида состоит в написании программ и описании структур и данных на специальных формальных языках программирования. Этап разработки и ввода программ в настоящее время автоматизирован благодаря использованию развивающихся многофункциональных систем программирования. С их помощью существенно облегчаются процесс создания программ, их отладка и ввод. Тем не менее, сам процесс моделирования, т.е. разработки моделей предметной области решаемых задач и их алгоритмической реализации, остается творческим и на этапе разработки информационных технологий в своей основе практически неавтоматизируем.

Таким образом, после сбора, подготовки, контроля и ввода исходная информация (документы, модели, программы) превращается в данные, представленные машинными (двоичными) кодами, которые хранятся на машинных носителях и обрабатываются техническими средствами информационной технологии.

3.1.3 Передача информации

Информационный процесс обмена предполагает обмен данными между процессами информационной технологии. Из рисунка 1 видно, что процесс обмена связан взаимными потоками данных со всеми информационными процессами на уровне переработки данных.

Передача информации осуществляется различными способами: с помощью курьера, пересылка по почте, доставка транспортными средствами, дистанционная передача по каналам связи, с помощью других средств коммуникаций.

При дистанционной передаче по каналам связи (рис. 3.3) можно выделить два основных типа процедур. Это процедуры передачи данных по каналам связи и сетевые процедуры, позволяющие осуществить организацию вычислительной сети. Процедуры передачи данных реализуются с помощью операции кодирования - декодирования, модуляции - демодуляции, согласования и усиления сигналов. Процедуры организации сети включают в себя в качестве основных операции по коммутации и маршрутизации потоков данных (трафика) в вычислительной сети. Процесс обмена позволяет, с одной стороны, передавать данные между источником и получателем информации, а с другой - объединять информацию многих ее источников.

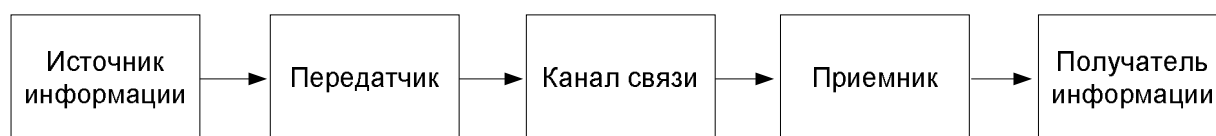


Рис. 3.3. Передача данных по каналу связи

Дистанционная передача по каналам связи сокращает время передачи данных, однако, для ее осуществления необходимы специальные технические средства, что удорожает процесс передачи. Предпочтительным является использование технических средств сбора и регистрации, которые, собирая автоматически информацию с установленных на рабочих местах датчиков, передают ее в ЭВМ для последующей обработки, что повышает ее достоверность и снижает трудоемкость.

Дистанционно может передаваться как первичная информация с мест ее возникновения, так и результатная в обратном направлении. В этом случае результатная информация фиксируется различными устройствами: дисплеями, табло, печатающими устройствами. Поступление информации по каналам связи в центр обработки в основном осуществляется двумя способами: на машинном носителе или непосредственно вводом в ЭВМ при помощи специальных программных и аппаратных средств. Дистанционная передача информации с помощью современных коммуникационных средств постоянно развивается и совершенствуется. Особое значение этот способ передачи информации имеет в многоуровневых межотраслевых системах, где применение дистанционной передачи значительно ускоряет прохождение

информации с одного уровня управления на другой и сокращает общее время обработки данных.

Модель обмена данными включает в себя формальное описание процедур, выполняемых в вычислительной сети: передачи, маршрутизации, коммутации. Именно эти процедуры и составляют информационный процесс обмена. Для качественной работы сети необходимы формальные соглашения между ее пользователями, что реализуется в виде протоколов сетевого обмена. В свою очередь, передача данных основывается на моделях кодирования, модуляции, каналов связи. На основе моделей обмена производится синтез системы обмена данными, при котором оптимизируются топология и структура вычислительной сети, метод коммутации, протоколы и процедуры доступа, адресации и маршрутизации.

В подсистему обмена данными входят комплексы программ и устройств, позволяющих реализовать вычислительную сеть и осуществить по ней передачу и прием сообщений с необходимыми скоростью и качеством.

Физическими компонентами подсистемы обмена служат устройства приема - передачи (модемы, усилители, коммутаторы, кабели, специальные вычислительные комплексы, осуществляющие коммутацию, маршрутизацию и Доступ к сетям). Программными компонентами подсистемы являются программы сетевого обмена, реализующие сетевые протоколы, кодирование - декодирование сообщений и др.

3.1.4 Обработка информации

Обработка информации производится на ПЭВМ, как правило, децентрализованно, в местах возникновения первичной информации, где организуются автоматизированные рабочие места специалистов той или иной управленческой службы (отдела материально-технического снабжения и сбыта, отдела главного технолога, конструкторского отдела, бухгалтерии, планового отдела и т.п.). Обработка, однако, может производиться не только автономно, но и в вычислительных сетях, с использованием набора ПЭВМ программных средств и информационных массивов для решения функциональных задач.

Процесс обработки данных связан с преобразованием данных и их отображением.

Процедуры преобразования данных на логическом уровне представляют собой алгоритмы и программы обработки данных и их структур. Сюда включаются стандартные процедуры, такие, как сортировка, поиск, создание и преобразование статистических и динамических структур данных, а также нестандартные процедуры, обусловленные алгоритмами и программами преобразования данных при решении конкретных информационных задач. Моделями процедур отображения данных являются компьютерные программы преобразования данных, представленных машинными кодами, в воспринимаемую человеком информацию, несущую в себе смысловое содержание. В современных ЭВМ данные могут быть отражены в виде

текстовой информации, в виде графиков, изображений, звука, с использованием средств мультимедиа, которые интегрируют в компьютере все основные способы отображения.

Процедура отображения данных - одна из важнейших в информационной технологии. Без возможности восприятия результата обработки информации человеческими органами чувств этот результат оставался бы вещью в себе (ведь мы не ощущаем машинное представление информации).

В современных информационных технологиях *при воспроизведении информации* предпочтение отдано графическим режимам работы дисплеев (в отличие от исторически более ранних текстовых режимов) как наиболее универсальным. Графический режим позволяет выводить на экран дисплея любую графику (ведь буквы и цифры тоже графические объекты), причем с возможностью изменения масштаба, проекции, цвета и т.д. Развитие информационных технологий относительно ввода и вывода информации идет по пути создания объектно-ориентированных систем, в которых настройка систем, программирование функциональных задач, ввод и вывод информации осуществляются с помощью графических объектов, отображаемых на экране дисплея (примером могут служить широко распространенный графический интерфейс Windows, объектно-ориентированные языки Delphi, Java и т.д.).

Отображение информации на экране дисплея в виде графических объектов (графиков, геометрических фигур, изображений и т. д.) носит название компьютерной графики.

В ходе решения задач на ЭВМ в соответствии с машинной программой формируется результатная информация, которая в дальнейшем используется на этапе принятия решения.

Принятие решения в автоматизированной системе организационного управления, как правило, осуществляется специалистом с применением или без применения технических средств, но в последнем случае на основе тщательного анализа результатной информации, полученной на ЭВМ. Задача принятия решений осложняется тем, что специалисту приходится искать из множества допустимых решений наиболее приемлемое, сводящее к минимуму потери ресурсов (временных, трудовых, материальных и т.д.). Благодаря применению персональных ЭВМ и терминальных устройств повышается аналитичность обрабатываемых сведений, а также обеспечивается постепенный переход к автоматизации выработки оптимальных решений в процессе диалога пользователя с вычислительной системой. Этому способствует использование новых технологий экспертных систем поддержки принятия решений.

Модель обработки данных включает в себя формализованное описание процедур организации вычислительного процесса, преобразования данных и отображения данных. Под организацией вычислительного процесса понимается управление ресурсами компьютера (память, процессор, внешние устройства) при решении задач обработки данных. Эта процедура формализуется в виде алгоритмов и программ системного управления

компьютером. Комплексы таких алгоритмов и программ получили название операционных систем.

Операционная система (ОС) - комплекс программ, организующих вычислительный процесс в вычислительной системе.

Вычислительная система - совокупность аппаратных и программных средств ЭВМ, взаимодействующих для решения задач обработки информации.

Для выполнения функций подсистемы обработки данных используются электронные вычислительные машины различных классов. В настоящее время при создании автоматизированных информационных технологий применяются три основных класса ЭВМ: на верхнем уровне - большие универсальные ЭВМ (по зарубежной классификации - мэйнфреймы), способные накапливать и обрабатывать громадные объемы исходных данных и используемые как главные ЭВМ; на среднем - абонентские вычислительные машины (серверы); на нижнем уровне - персональные компьютеры либо управляющие ЭВМ. Обработка данных, т.е. их преобразование и отображение, производится с помощью программ решения задач в той предметной области, для которой создана информационная технология.

3.1.5 Хранение и накопление информации

Хранение и накопление информации вызвано многократным ее использованием, применением условно-постоянной, справочной и других видов информации, необходимостью комплектации первичных данных до их обработки.

Назначение технологического процесса накопления данных состоит в создании, хранении и поддержании в актуальном состоянии информационного фонда, необходимого для выполнения функциональных задач системы управления. Хранение и накопление информации осуществляется в информационных базах в виде информационных массивов, где данные располагаются по установленному в процессе проектирования порядку.

С хранением и накоплением непосредственно связан поиск данных, т. е. выборка нужных данных из хранимой информации, включая поиск информации, подлежащей корректировке или замене. Процедура поиска информации выполняется автоматически на основе составленного пользователем или ПЭВМ запроса на нужную информацию.

Указанные функции, выполняемые в процессе накопления данных, реализуются по алгоритмам, разработанным на основе соответствующих математических моделей.

Процесс накопления данных состоит из ряда основных процедур, таких, как выбор хранимых данных, хранение данных, их актуализация и извлечение.

Информационный фонд систем управления должен формироваться на основе принципов необходимой полноты и минимальной избыточности хранимой информации. Эти принципы реализуются процедурой выбора хранимых данных, в процессе выполнения которой производится анализ циркулирующих в системе данных и на основе их группировки на входные,

промежуточные и выходные определяется состав хранимых данных. Входные данные - это данные, получаемые из первичной информации и создающие информационный образ предметной области. Они подлежат хранению в первую очередь. Промежуточные данные - это данные, формирующиеся из других данных при алгоритмических преобразованиях. Как правило, они не хранятся, но накладывают ограничения на емкость оперативной памяти компьютера. Выходные данные являются результатом обработки первичных (входных) данных по соответствующей модели, они входят в состав управляющего информационного потока своего уровня и подлежат хранению в определенном временном интервале. Вообще, данные имеют свой жизненный цикл существования, который фактически и отображается в процедурах процесса накопления.

Процедура хранения состоит в том, чтобы сформировать и поддерживать структуру хранения данных в памяти ЭВМ. Современные структуры хранения данных должны быть независимы от программ, использующих эти данные, и реализовывать вышеуказанные принципы (полнота и минимальная избыточность). Такие структуры получили название баз данных. Осуществление процедур создания структуры хранения (базы данных), актуализации, извлечения и удаления данных производится с помощью специальных программ, называемых системами управления базами данных.

В процессе накопления данных важной *процедурой является их актуализация*. Под актуализацией понимается поддержание хранимых данных на уровне, соответствующем информационным потребностям решаемых задач в системе, где организована информационная технология. Актуализация данных осуществляется с помощью операций добавления новых данных к уже хранимым, корректировки (изменения значений или элементов структур) данных и их уничтожения, если данные устарели и уже не могут быть использованы при решении функциональных задач системы.

Процедура извлечения данных из базы необходима для пересылки требуемых данных либо для преобразования, либо для отображения, либо для передачи по вычислительной сети.

При выполнении процедур актуализации и извлечения обязательно выполняются операции поиска данных по заданным признакам и их сортировки, состоящие в изменении порядка расположения данных при хранении или извлечении.

На логическом уровне все процедуры процесса накопления должны быть формализованы, что отображается в математических и алгоритмических моделях этих процедур.

Модель накопления данных формализует описание информационной базы, которая в компьютерном виде представляется базой данных.

Процесс перехода от информационного (смыслового) уровня к физическому описывается трехуровневой системой моделей представления информационной базы: *концептуальной, логической и физической схем*.

Концептуальная схема информационной базы описывает информационное содержание предметной области, т.е. какая и в каком объеме

информация должна накапливаться при реализации информационной технологии.

Логическая схема информационной базы должна формализовано описать ее структуру и взаимосвязь элементов информации. При этом могут быть использованы различные подходы: реляционный, иерархический, сетевой. Выбор подхода определяет и систему управления базой данных, которая, в свою очередь, определяет физическую модель данных - физическую схему информационной базы, описывающую методы размещения данных и доступа к ним на машинных (физических) носителях информации. Модель данных - формализованное описание информационных структур и операций над ними.

3.2 Системный подход к решению функциональных задач и к организации информационных процессов

Пока автоматизация решения задач управления носила локальный, частный характер, а количество таких задач было невелико, схема технологического процесса создания программных средств, при которой каждая функциональная задача рассматривалась отдельно от других задач, могла в большей или меньшей степени удовлетворять разработчиков. Когда возникла потребность создания систем автоматизированной обработки информации, внедрение которых могло обеспечить совершенствование организационно-экономического управления, такая схема оказалась непригодной, так как она не отражала основного принципа разработки - принципа системного подхода, что проявилось особенно ярко в виде массового дублирования данных в информационных массивах.

В качестве альтернативы такому дублированию информации возникла концепция баз данных как единого, централизованного хранилища всей информации, необходимой для решения задач управления. Первоначально в противовес огромному дублированию информации, присущему позадачному подходу, концепция БД подразумевала полное отсутствие такого дублирования. Однако теоретически корректная концепция в реальности оказалась малоэффективной, так как безусловный выигрыш в объемах необходимой памяти оборачивался значительным проигрышем во времени, требуемым на поиск и выборку из БД информации, необходимой для решения той или иной конкретной задачи.

В связи с этим в настоящее время концепция БД подразумевает разумный компромисс между сокращением до минимума необходимого дублирования информации и эффективностью процесса выборки и обновления требуемых данных. Действительное обеспечение такого решения возможно только при условии системного анализа всего комплекса задач, подлежащих автоматизации, уже на этапе описания системы: ее целей и функций, состава и специфики информационных потоков, информационного состава задач и даже отдельных программных модулей. Системный подход, базирующийся на положениях общей теории систем, наиболее эффективен

при решении сложных задач анализа и синтеза, требующих одновременного использования ряда научных дисциплин.

Другим важным фактором, обуславливающим необходимость системного подхода, начиная с этапа формулирования требований и постановки задач, является то, что на этот этап приходится до 70-80% всех затрат, падающих на разработку прикладного программного обеспечения, и он имеет особое значение в обеспечении соответствия результатов разработки потребностям конечных пользователей.

Последнее особенно важно, так как по тяжести последствий ошибок этот этап занимает первое место среди всех остальных этапов.

Так, по проведенному статистическому анализу большого числа проектов, выполненных ведущими западными компьютерными фирмами (IBM, TRW, GTE Corp., Bell Labs.), в типовом программистском проекте 56% всех обнаруженных ошибок приходится на ошибки в требованиях на программы, а на устранение этих ошибок уходит до 82% всех усилий, затрачиваемых разработчиками на устранение ошибок проекта. Такое положение дел объясняется, с одной стороны, сложностью и трудоемкостью этапа в плане обеспечения адекватности трактовки разработчиками проекта требований пользователей, а с другой стороны, тем, что неизбежные ошибки, допущенные на этом этапе, как правило, обнаруживаются (проявляются) лишь на стадии опытной и даже промышленной эксплуатации, когда стоимость их исправления возрастает в десятки раз.

Объективное требование системного подхода к разработке программных средств решения задач при автоматизации систем управления вызвало необходимость дифференциации специалистов-разработчиков, что проявилось в выделении в их составе системных аналитиков, системотехников, прикладных и системных программистов.

Системный аналитик, исходя из общих целей, назначения, технических характеристик, состава и описания требований пользователей к прикладным задачам и системе в целом, формулирует общие формальные требования к программному обеспечению системы.

Специалист-системотехник преобразует общие формальные требования в детальные спецификации на отдельные программы, участвует в разработке логической структуры базы данных и т.п., т.е. определяет общую информационно-программную структуру проекта.

Прикладной программист преобразует спецификации в логическую структуру программных модулей, а затем и в программный код.

Системный программист обеспечивает сопряжение программных модулей с программной средой, в рамках которой предстоит функционировать прикладным программам (задачам).

В целях сокращения общей длительности разработки системы начало некоторых этапов технологического процесса осуществляется еще до полного завершения работ на предыдущем этапе. Такой частичный параллелизм в работе, кроме того, обуславливается и итерационным характером работ на этих этапах, когда в ходе выполнения отдельных работ 1-го этапа возникает

необходимость уточнения или изменения спецификаций, выполненных на предшествующих этапах, либо пользователь по своей инициативе вносит коррективы в исходные требования, что, естественно, отражается на всей последующей технологической цепочке реализации проекта.

Другой отличительной чертой системной разработки проектов прикладных программ является их ориентация на использование интегрированных и распределенных баз данных. В связи с этим в качестве инструментальных средств разработки компонентов программного обеспечения наряду с языками программирования стали применяться языковые средства СУБД.

Микропроцессорная революция резко поменяла приоритеты и актуальность проблем, присущих традиционным технологиям разработки прикладных программ. Быстро растущая вычислительная мощность, расширение других вычислительных возможностей современных ПК в сочетании с возможностью объединения этих ресурсов с помощью вычислительных сетей - все это позволило нивелировать погрешности Пользователей - непрофессиональных программистов в плане эффективности создаваемых ими программных средств решения прикладных задач.

Возможность исключения из технологической цепочки программистов-профессионалов (посредников) создала предпосылки для ускорения процесса разработки прикладных программных средств, а главное для сокращения количества ошибок, присущих традиционным технологическим схемам, когда основные усилия профессиональных программистов затрачивались на то, чтобы адекватно воспринять требования, предъявляемые конечными пользователями к программам, обеспечить своевременное получение достоверных, исчерпывающих сведений об исходных данных, необходимых для решения задачи, и т.п.

Но эффект от такого "вытеснения" профессиональных программистов из их сферы деятельности пользователями-непрофессионалами зачастую снижался или не ощущался вообще в связи с тем, что, не владея основами методологии разработки программных средств, типовыми программистскими приемами и умением использовать "подручные" средства из арсенала той или иной инструментальной среды, пользователи зачастую попадают в различные "тупиковые" ситуации, которые не составляют каких-либо трудностей для профессионалов в области программирования.

Раздел 4. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В РАЗЛИЧНЫХ ОБЛАСТЯХ ДЕЯТЕЛЬНОСТИ

4.1 Информационные технологии в системах организационного управления

5.3.1 ЭВМ при выборе решений в области технологии, организации, планирования и управления производством

Применение компьютерных информационных технологий позволяет в ряде случаев при сравнительно небольших затратах получать ценные управленческие решения. Составление экономико-математических моделей и проведение расчетов с помощью компьютера позволяют быстро и относительно недорого проводить разработку и сравнение многочисленных вариантов планов и управленческих решений.

Многовариантность выбора - одно из ценнейших качеств рассматриваемых методов. Однако в настоящее время практическое применение экономико-математических методов в управление и планировании производственной деятельностью, несмотря на оснащение управленческих служб средствами вычислительной техники, далеко не соответствует имеющемуся в этой области научному запасу.

Трудности практического внедрения экономико-математических методов связаны со многими объективными и субъективными причинами, но прежде всего обусловлены сложностью экономических процессов и явлений, невозможностью расчленения больших систем на обозримые части с целью их автономного рассмотрения, а также *необходимостью* учитывать наряду с технологическими аспектами и поведение людей.

Поэтому практически приемлемым путем является включение компьютерных решений конкретных типовых задач в процесс принятия управленческих решений руководителем. При этом необходимо сочетать опыт и трудноформализуемые знания руководителя, хорошо знающего производственную и хозяйственную стороны управленческой деятельности, с производительностью и многовариантностью компьютерно-математических методов.

В настоящее время имеются отработанные методы решения ряда типовых задач по организации и планированию производства, для которых могут быть применены компьютерные технологии. Все эти задачи могут быть классифицированы следующим образом.

1) Задачи в области организации производства. К ним относятся, например, задачи организации проектирования, ремонта машин, транспорта и складского хозяйства, задачи управления качеством, расчета потребности в ресурсах (трудовых, материальных, технических) с распределением во времени на основе календарного плана производства и т.п.

2) Задачи планирования производства. К ним относятся, например, задачи планирования производства товарной продукции, технического развития

и повышения эффективности производства, труда и заработной платы, механизации и материально-технического обеспечения производства, задачи анализа производственно-хозяйственной деятельности и т. п.

Такие отработанные решения определенных типовых задач базируются на методах имитационного моделирования, линейного программирования, вероятностного моделирования и других методах.

Возможность практического решения указанных задач в настоящее время расширяется в связи с компьютеризацией всех звеньев управленческого аппарата, созданием локальных и объединенных вычислительных сетей, организацией локальных и централизованных информационных баз данных и обеспечением к ним оперативного доступа.

5.3.2 Возможности использования новых информационных технологий в системах организационного управления

Современные информационные технологии определяются как непрерывные процессы обработки, хранения, передачи и отображения информации, направленные на эффективное использование информационных ресурсов, средств вычислительной техники и передачи данных при управлении системами различного класса и назначения.

ИТ существенно увеличивают степень автоматизации всех информационных процессов, что является предпосылкой для ускорения темпов научно-технического прогресса, повышения производительности и эффективности управленческого труда.

Основу современных информационных технологий составляют *«четыре»* технических достижения:

- 1) развитие носителей информации, позволяющих хранить практически неограниченные объемы информации;
- 2) развитие средств связи, обеспечивающих доставку информации в любую точку земного шара без существенных ограничений во времени;
- 3) возможность автоматизированной обработки информации в местах ее возникновения с помощью персональной ЭВМ;
- 4) возможности удаленного доступа и обработки информации, хранящейся в распределенных базах и банках данных.

ИТ развивались в процессе целенаправленной интеграции средств хранения, обработки, передачи и представления информации в комплексные системы, обеспечивающие циркуляцию требуемых потоков данных в рамках определенных организационных систем.

На современном этапе автоматизированная обработка данных в организационных системах характеризуется переходом от централизованной обработки информации к распределенной (децентрализованной), на основе широкого применения персональных ЭВМ.

Объединение ЭВМ в сети (локальные и региональные) позволяет пользователям сочетать преимущества автономной распределенной обработки

информации с возможностями индивидуального доступа к общим информационным ресурсам отдела, предприятия, района и т.д.

Ввод и обработка информации на рабочем месте сотрудника (руководителя и специалиста) с использованием ПК позволяет повысить качество, точность, своевременность и актуальность подготавливаемых документов и увеличить скорость их подготовки.

Объединение автоматизированных рабочих мест сотрудников в локальные вычислительные сети (ЛВС) позволяет снизить затраты на информационный обмен, решить задачу оптимального использования вычислительных мощностей и ресурсов. Включение в качестве элемента ЛВС высокопроизводительной ЭВМ с внешними запоминающими устройствами большого объема позволяет централизовать информацию, необходимую для совместной обработки всеми пользователями сети и исключить дублирование такой информации.

Технические средства "электронной почты" и вычислительных систем позволяют внедрять в организационных системах безбумажные технологии, при которых часть информационных потоков и массивов (файлов) переносятся на бумажный носитель лишь в строго регламентированных случаях, связанных в основном с подготовкой и представлением в официальных итоговых отчетов.

В современном учреждении выполняется несколько десятков видов работ, включающих:

- осуществление информационных коммуникаций внутри организации и между организациями;
- изучение, поиск, накопление и генерирование информации (чтение документов, подготовка отчетов, писем, ответов на письма, поиск необходимых данных, ведение архивов и т.п.);
- анализ данных и принятие решений;
- управление функционированием организации;
- информационное обслуживание руководителей и т. д.

Основными элементами современного "электронного" учреждения являются автоматизированные рабочие места (АРМы) пользователей, системы редактирования текстов, базы данных и средства управления ими (СУБД), информационно-вычислительные сети, электронная почта, средства печати и копирования документов и др.

Автоматизированное рабочее место (АРМ) - вычислительная система, предназначенная для автоматизации профессиональной деятельности.

Производительность труда при использовании АРМ на рутинных операциях, применяемых при подготовке и передаче документов увеличивается в несколько раз за счет применения специального программного обеспечения.

Примеры функций пользователей - сотрудников, реализуемые на АРМе соответствующего типа являются:

- подготовка документов, содержащих текстовые, табличные и графические фрагменты на основе анализа доступной информации;
- хранение и поиск информации;

- прием/передача документов (или их фрагментов) внутри учреждения и за его пределы;
- обеспечение режима использования и надежного хранения документов.

Функции АРМов руководителей организации и ее подразделений существенно отличаются от функций АРМ сотрудника (служащего, специалиста).

К основным функциям руководителя относятся: долгосрочное и оперативное планирование работ, общение со смежными подразделениями, проведение рабочих совещаний, выдача поручений и контроль за их выполнением, регистрация и исполнение поручений руководства, оценка деятельности сотрудников, подразделения и организации в целом и другие функции. Большинство этих функций может быть успешно реализовано при наличии соответствующего прикладного программного обеспечения АРМ руководителя.

Таким образом, внедрение информационных технологий в процесс управления организациями не ограничивается только автоматизацией сбора, хранения и представления данных, а распространяется также на анализ информации и поддержку принятия решений. В большинстве случаев решения принимаются на основе математического моделирования технико-экономической ситуации в конкретной предметной области. Рассматриваемое в таком аспекте АРМ руководителя становится усилителем его интеллекта, помогает находить достаточно эффективные (неубыточные) управленческие решения в сложных, динамически изменяющихся ситуациях.

Необходимость в обмене информацией в различных сферах управленческой деятельности, получении новых сведений в результате коллективного обсуждения проблем привели к таким формам общения, как конференции, семинары, совещания.

Практически ни одна серьезная сделка, ни один договор не могут быть заключены без обсуждения на различных уровнях промежуточных результатов, итогов, вариантов решения, заслушивания оппонентов и принятия соответствующих решений. Эта идея была реализована новом виде информационного обслуживания - *телеконференции*. Участники таких конференций, удаленные друг от друга на сотни и тысячи километров, благодаря современной электронике могут видеть друг друга, обмениваться данными и графической информацией дискутировать в условиях, максимально приближенных к реальной конференции.

Организация телеконференций требует привлечения весьма разнообразной аппаратуры: терминалов, факсимильной связи, телевизионных камер, видеомagneтофонов, компьютеров, модемов, акустической аппаратуры.

Новейшей технологией информационного обслуживания организаций являются создание использование *автоматизированных информационных систем интеллектуального типа* (экспертные системы, функционирующие в режиме реального времени, ситуационные комнаты, интеллектуальные системы поддержки принятия решения и т.д.).

4.2 Информационные технологии в обучении

Создание и совершенствование компьютеров привело и продолжает приводить к созданию новых технологий в различных сферах научной и практической деятельности. Одной из таких сфер стало образование. Нетрадиционные информационные системы, связанные с обучением, называют информационно-обучающими.

Автоматизированная обучающая система (АОС) - комплекс программных, технических и учебно-методических средств, предназначенных для активного индивидуального обучения человека на основе программного управления этим обучением.

Благодаря своим конструктивным и функциональным особенностям современный персональный компьютер находит применение в обучении самым разнообразным дисциплинам и служит базой для создания большого числа новых информационных технологий обучения.

Компьютерная технология повышает интерес к обучению. В настоящее время существует огромное множество обучающих программ по самым разным предметам, ориентированных на самые различные категории учащихся, начиная с детских садов и заканчивая персоналом атомных электростанций.

Типы обучающих программ

Основанием для классификации служат обычно особенности учебной деятельности обучаемых при работе с программами. Обычно выделяют четыре типа обучающих программ:

- тренировочные и контролирующие;
- наставнические;
- имитационные и моделирующие;
- развивающие игры.

Тренировочные программы предназначены для закрепления умений и навыков. Предполагается, что теоретический материал уже изучен. Эти программы в случайной последовательности предлагают учащемуся вопросы и задачи и подсчитывают количество правильно и неправильно решенных задач (в случае правильного ответа может выдаваться поощряющая реплика, при неправильном ответе можно получить помощь в виде подсказки).

Наставнические программы предлагают ученикам теоретический материал для изучения. Задачи и вопросы служат в этих программах для организации человеко-машинного диалога, для управления ходом обучения. Так, если ответы, даваемые учеником, неверны, программа может "откатиться назад" для повторного изучения теоретического материала.

Программы наставнического типа являются прямыми наследниками средств программированного обучения 60-х годов в том смысле, что основным теоретическим источником современного компьютерного или автоматизированного обучения следует считать программированное обучение. В публикациях зарубежных специалистов и сегодня под термином "программированное обучение" понимают современные компьютерные

технологии. Одним из основоположников концепции программированного обучения является американский психолог Б.Ф.Скиннер.

Главным элементом программированного обучения является программа, понимаемая как упорядоченная последовательность рекомендаций (задач), которые передаются с помощью программированного учебника и выполняются обучаемыми. Существует несколько разновидностей программированного обучения.

- линейное программированное обучение. Линейная программа характеризуется следующими особенностями:
- разветвленная программа. Разветвленная программа основана выборе одного правильного ответа из нескольких данных, она

Если основой линейной программы является стремление избежать ошибок, то разветвленная программа не направлена на ликвидацию ошибок в процессе обучения: ошибки трактуются как возможность обнаружить недостатки в знаниях обучаемых, а также выяснить, какие проблемы обучаемые уяснили недостаточно. Постепенно (линейное и разветвленное программированное) уступили место смешанным формам.

Существует и продолжает разрабатываться большое количество *инструментальных* программ такого вида. Общим их недостатком является высокая трудоемкость разработки, затруднения организационного и методического характера при использовании в реальном процессе обучения.

Моделирующие программы основаны на графических иллюстративных возможностях компьютера, с одной стороны, и вычислительных, с другой, и позволяют осуществлять *компьютерный эксперимент*. Такие программы предоставляют возможность наблюдать на экране дисплея некоторый процесс, влияя на его ход подачей команды с клавиатуры, меняющей значения параметров.

Развивающие игры предоставляют в распоряжение ученика некоторую воображаемую среду, существующий только в компьютере мир, набор каких-то возможностей и средств их реализации. Использование предоставляемых программой средств для реализации возможностей, связанных с изучением мира игры и деятельностью в этом мире, приводит к развитию обучаемого, формированию у него познавательных навыков, самостоятельному открытию им закономерностей, отношений объектов действительности, имеющих значение.

Наибольшее распространение получили обучающие программы первых двух типов в связи с их относительно невысокой сложностью, возможностью унификации при разработке многих блоков программ. Если программы 3-го и 4-го типов требуют большой работы программистов, психологов, специалистов в области изучаемого предмета, педагогов-методистов, то технология создания программ 1-го и 2-го типов сегодня сильно упростилась с появлением инструментальных средств или наполняемых автоматизированных обучающих систем.

В процессе контроля знаний широкое распространение получило компьютерное тестирование. В ряде стран тестирование вытеснило

традиционные формы контроля - устные и письменные экзамены и собеседования.

Типы компьютерных тестовых заданий определяются способами однозначного распознавания ответных действий тестируемого в соответствии с моделью знаний.

Учебная мультимедиа и гипермедиа-технология представляет собой развитие технологии программированного обучения, хотя упор делается не на адаптивность обучения и его методическое обоснование, а на внешнюю иллюстративно-наглядную сторону. Современные графические и звуковые возможности компьютера обусловили появление средств гипер- и мультимедиа.

Мультимедиа технология - представление информации в форме видеоизображения с применением мультипликации и звукового сопровождения.

Гипермедиа технология - компьютерное представление данных различного типа, в котором автоматически поддерживаются смысловые связи между выделенными понятиями, объектами или разделами.

Научные исследования в данной области связаны с разработкой технологий создания учебных курсов большего размера на основе возможностей мульти- и гипермедиа. Под управлением компьютера система мультисред может производить в едином представлении объединение текста, графики, звуков, видеообразов и мультипликации. Технология мультимедиа в последнее время широко применяется для создания электронных книг и учебников.

Развитием идей мультимедиа являются технологии компьютерной виртуальной реальности. В этом случае с помощью специальных экранов, датчиков, шлемов, перчаток и т.п. полностью моделируется управление, например, самолетом, так что у обучаемого возникает полная иллюзия того, что он находится в кабине самолета и им управляет.

4.3 Автоматизированные системы научных исследований

Автоматизированные системы научных исследований (АСНИ) представляют собой программно-аппаратные комплексы, обрабатывающие данные, поступающие от различного рода экспериментальных установок и измерительных приборов, и на основе их анализа облегчающие обнаружение новых эффектов и закономерностей (рис. 1).

Блок связи с измерительной аппаратурой преобразует к нужному виду информацию, поступающую от измерительной аппаратуры.

В базе данных хранится информация, поступившая из блока связи с измерительной аппаратурой, а также заранее введенная с целью обеспечения работоспособности системы.

Расчетный блок, выполняя программы из пакета прикладных программ, производит все математические расчеты, в которых может возникнуть потребность в ходе научных исследований.

Расчеты могут выполняться как по требованию исследователя, так и блока имитационного моделирования. При этом на основе математических моделей воспроизводится процесс, происходящий во внешней среде.

Экспертная система моделирует рассуждения специалистов данной предметной области. С ее помощью исследователь может классифицировать наблюдаемые явления, диагностировать течение следуемых процессов.

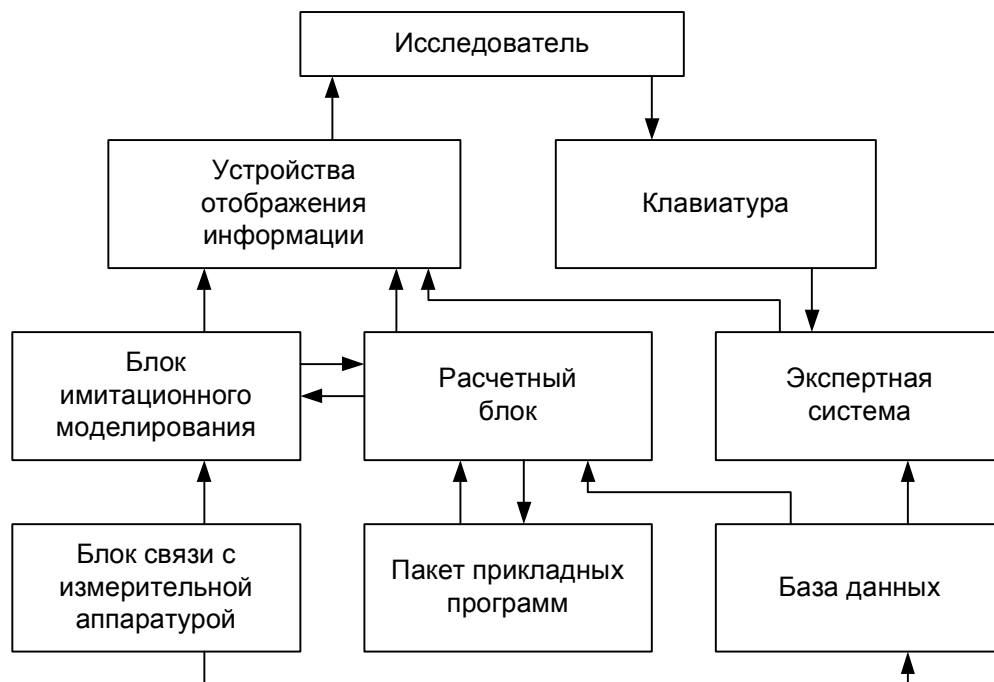


Рис. 4.1 - Типовая структура АСНИ

АСНИ получили широкое распространение в молекулярной химии, минералогии, биохимии, физике элементарных частиц и многих других науках.

4.4 Системы автоматизированного проектирования

Близкими по своей структуре и функциям к системам автоматизации научных исследований оказываются системы автоматизированного проектирования (САПР).

САПР - комплекс программных и аппаратных средств, предназначенных для автоматизации процесса проектирования человеком технических изделий или продуктов интеллектуальной деятельности.

Проектирование новых изделий - основная задача изобретателей конструкторов, протекает в несколько этапов, таких как нормирование замысла, поиск физических принципов, обеспечивающих реализацию замыслов и требуемые значения конструкции, поиск конструктивных решений, их расчет и обоснование, создание опытного образца, разработка технологий промышленного изготовления. Если формирование замысла и поиск физических принципов пока остаются чисто творческими, не

поддающимися автоматизации этапами, то при конструировании и расчетах с успехом могут быть применены САПР (рис. 4.2).

База данных, блок имитационного моделирования, расчетный блок и экспертная система выполняют функции, аналогичные функциям соответствующих блоков АСНИ. Вместо блока связи с измерительной аппаратурой в САПР имеется блок формирования заданий. Проектировщик вводит в блок техническое задание на проектирование, в котором указаны цели, которые необходимо достичь при проектировании, и все ограничения, которые нельзя нарушить. Блок подготовки технической документации облегчает создание технической документации для последующего изготовления изделия.

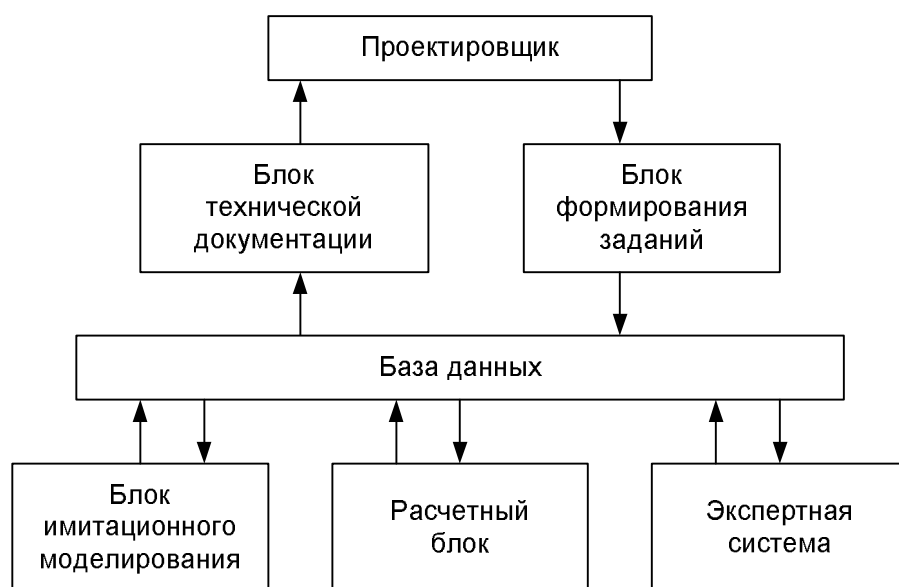


Рис 4.2 - Типовая схема САПР

Аппаратное обеспечение САПР составляет ЭВМ с набором устройств, необходимых для ввода и вывода графической информации (графопостроитель, световое перо, графический планшет и др.).

В настоящее время САПР является неотъемлемым атрибутом крупных конструкторских бюро и проектных организаций, работающих в различных предметных областях. Это важная сфера приложения идей и методов информатики. САПР широко применяется в архитектуре, электротехнике, электронике, машиностроении, авиакосмической технике и др.

4.5 Геоинформационные системы и технологии

Геоинформационные системы (ГИС) и ГИС- технологии объединяют компьютерную картографию и системы управления базами данных. Концепция технологии ГИС состоит в создании многослойной электронной карты, опорный слой которой описывает географию территории, а каждый из остальных слоев - один из аспектов состояния территории. Тем

самым *ГИС-технологии* определяют специфическую область работы с информацией.

Технология ГИС применима везде, где необходимо учитывать, обрабатывать и демонстрировать территориально распределенную информацию. Пользователями ГИС-технологии могут быть как организации, чья деятельность целиком базируется на земле владельцы нефтегазовых предприятий, экологические службы, жилищно-коммунальное хозяйство, так и многочисленные коммерческие предприятия - банки, страховые, торговые и строительные фирмы, чья успешная работа во многом зависит от правильного и своевременного учета территориального фактора.

В основе любой ГИС лежит информация о каком-либо участке земной поверхности: континенте, стране, городе, улице.

БД организуется в виде набора слоев информации. Основной шрифт содержит географически привязанную карту местности (топооснова). На него накладываются другие слои, несущие информацию об *объектах*, находящихся на данной территории: коммуникации, в том числе линии электропередач, нефте- и газопроводы, водопроводы, промышленные объекты, земельные участки, почвы, коммунальное хозяйство, землепользование и др.

В процессе создания и наложения слоев друг на друга между ними устанавливаются необходимые связи, что позволяет выполнять пространственные операции с объектами посредством моделирования и интеллектуальной обработки данных.

Как правило, информация представляется графически в векторном виде, что позволяет уменьшить объем хранимой информации и упростить операции по визуализации. С графической информацией связана текстовая, табличная, расчетная информация, координатная привязка к карте местности, видеоизображения, аудиокomentarии, БД с описанием объектов и их характеристик.

Многие ГИС включают аналитические функции, которые позволяют моделировать процессы, основываясь на картографической информации.

Программное ядро ГИС можно условно разделить на две подсистемы: СУБД и управление графическим выводом изображения. В качестве СУБД используют SQL-серверы.

Рассмотрим типовую схему организации ГИС-технологии, в настоящее время сложился основной набор компонентов, составляющих ГИС. К ним относятся:

- 1) приобретение и предварительная подготовка данных;
- 2) ввод и размещение данных;
- 3) управление данными;
- 4) манипуляция данными и их анализ;
- 5) производство конечного продукта.

Функциональным назначением данных компонентов является:

Приобретение и подготовка исходных данных; включает

манипуляции с исходными данными карт - материалами на твердой или бумажной основе, данными дистанционного зондирования, результатами полевых испытаний, текстовыми (табличными) материалами, с архивными данными.

Ввод и размещение пространственной и непространственной составляющих данных включает конвертирование информации во внутренние форматы системы и обеспечение структурной и логической совместимости всего множества порождаемых данных.

Управление данными предполагает наличие средств оптимальной внутренней организации данных, обеспечивающих эффективный доступ к ним.

Функции манипуляции и анализа представлены средствами, предназначенными для содержательной обработки данных в целях обработки и реорганизации данных. С точки зрения пользователя, эти функции являются главными в *ГИС-технологиях*, потому что позволяют получать новую информацию, необходимую для управления, исследовательских целей, прогнозирования.

Производство конечного продукта включает вывод полученных результатов для конечных потребителей ГИС. Эти продукты могут представлять карты, статистические отчеты, различные графики, стандартные формы определенных документов.

Кроме этого, каждый картографический объект может иметь атрибутивную информацию, в которой содержится информация, которая не обязательно должна отображаться на карте (например, число жильцов какого-либо дома и их социальный статус).

подавляющее большинство ГИС-систем различают геометрическую и атрибутивную компоненты баз данных ГИС. Их часто называют также пространственными (картографическими, геометрическими) и непространственными (табличными, реляционными) данными.

Картографическая информация представляется точками, кривыми и площадными объектами.

Атрибутивная информация содержит текстовые, числовые, логические данные о картографических объектах. Большинство современных ГИС-инструментариев позволяют хранить информацию в составе БД, как правило, реляционных.

Атрибутивная информация хранится в виде отдельных табличных файлов, как правило, в форматах реляционных баз данных систем DBF, PARADOX, ORACLE, INGRESS. Такой способ характерен как для западных коммерческих продуктов, так и современных отечественных разработок.

5.1 Технологии распределенных вычислений (РВ)

Современное производство требует высоких скоростей обработки информации, удобных форм ее хранения и передачи. Необходимо также иметь динамичные способы обращения к информации, способы поиска данных в заданные временные интервалы, чтобы реализовывать сложную математическую и логическую обработку данных.

Управление крупными предприятиями, управление экономикой на уровне страны требуют участия в этом процессе достаточно крупных коллективов. Такие коллективы могут располагаться в различных районах города, в различных регионах страны и даже в различных странах. Для решения задач управления, обеспечивающих реализацию экономической стратегии, становятся важными и актуальными скорость и удобство обмена информацией, а также возможность тесного взаимодействия всех участвующих в процессе выработки управленческих решений.

В эпоху централизованного использования ЭВМ с пакетной обработкой информации пользователи вычислительной техники предпочитали приобретать компьютеры, на которых можно было бы решать почти все классы их задач. Однако сложность решаемых задач обратно пропорциональна их количеству, и это приводило к неэффективному использованию вычислительной мощности ЭВМ при значительных материальных затратах. Нельзя не учитывать и тот факт, что доступ к ресурсам компьютеров был затруднен из-за существующей политики централизации вычислительных средств в одном месте.

Принцип централизованной обработки данных (рис. 5.1) не отвечал высоким требованиям к надежности процесса обработки, затруднял развитие систем и не мог обеспечить необходимые временные параметры при диалоговой обработке данных в многопользовательском режиме. Кратковременный выход из строя центральной ЭВМ приводил к роковым последствиям для системы в целом.

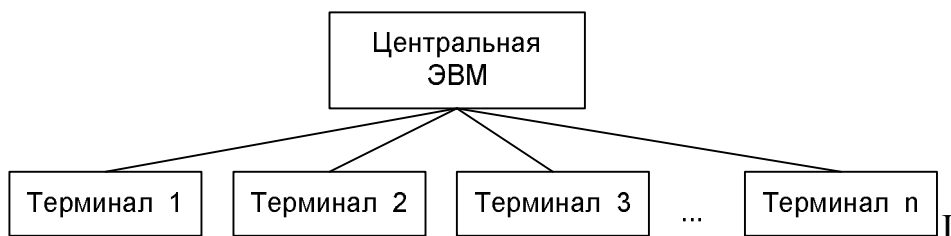


Рис. 5.1 - Система централизованной обработки данных

Появление персональных компьютеров потребовало нового подхода к организации систем обработки данных, к созданию новых информационных технологий. Возникло логически обоснованное требование перехода от

использования отдельных ЭВМ в системах централизованной обработки данных к распределенной обработке данных (рис. 5.2).

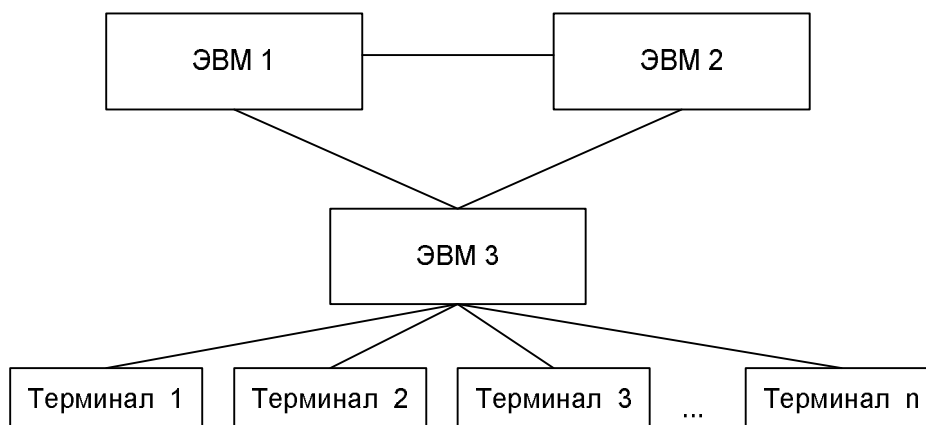


Рис. 5.2 - Система распределенной обработки данных

Распределенная обработка данных - обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

В основе распределенных вычислений лежат две основные идеи:

- много организационно и физически распределенных пользователей, одновременно работающих с общими данными - общей базой данных (пользователи с разными именами, которые могут располагаться на различных вычислительных установках, с различными полномочиями и задачами);
- логически и физически распределенные данные, составляющие и образующие тем не менее, общую базу данных (отдельные таблицы, записи и даже поля могут располагаться на различных вычислительных установках или входить в различные локальные базы данных).

Для реализации распределенной обработки данных были созданы многомашинные ассоциации, структура которых разрабатывается по одному из следующих направлений:

- многомашинные вычислительные комплексы (МВК);
- компьютерные (вычислительные) сети.

Многомашинный вычислительный комплекс - группа установленных рядом вычислительных машин, объединенных с помощью специальных средств сопряжения и выполняющих совместно единый информационно-вычислительный процесс. Под процессом понимается некоторая последовательность действий для решения задачи, определяемая программой.

Многомашинные вычислительные комплексы могут быть:

- локальными, при условии установки компьютеров в одном помещении, не требующих для взаимосвязи специального оборудования и каналов связи;

- дистанционными, если некоторые компьютеры комплекса установлены на значительном расстоянии от центральной ЭВМ и для передачи данных используются телефонные каналы связи.

Пример 1. Три ЭВМ объединены в комплекс для распределения заданий, поступающих на обработку. Одна из них выполняет диспетчерскую функцию и распределяет задания в зависимости от занятости одной из двух других обрабатывающих ЭВМ. Это локальный многомашинный комплекс.

Пример 2. ЭВМ, осуществляющая сбор данных по некоторому региону, выполняет их предварительную обработку и передает для дальнейшего использования на центральную ЭВМ по телефонному каналу связи. Это дистанционный многомашинный комплекс.

Компьютерная (вычислительная) сеть - вычислительная система, включающая в себя несколько компьютеров, терминалов и других аппаратных средств, соединенных между собой линиями связи, обеспечивающими передачу данных

Терминал - устройство, предназначенное для взаимодействия пользователя с вычислительной системой или сетью ЭВМ. Состоит из устройства ввода (чаще всего это клавиатура) и одного или нескольких устройств вывода (дисплей, принтер и т.д.).

5.2 Распределенные базы данных

Системы распределенных вычислений появляются, прежде всего, по той причине, что в крупных автоматизированных информационных системах, построенных на основе корпоративных сетей, не всегда удается организовать централизованное размещение всех баз данных и СУБД на одном узле сети. Поэтому системы распределенных вычислений тесно связаны с системами управления распределенными базами данных.

Распределенная база данных - это совокупность логически взаимосвязанных баз данных, распределенных в компьютерной сети.

Система управления распределенной базой данных - это программная система, которая обеспечивает управление распределенной базой данных и прозрачность ее распределенности для пользователей.

Распределенная база данных может объединять базы данных, поддерживающие любые модели (иерархические, сетевые, реляционные и объектно-ориентированные базы данных) в рамках единой глобальной схемы. Подобная конфигурация должна обеспечивать для всех приложений прозрачный доступ к любым данным независимо от их местоположения и формата.

Основные принципы создания и функционирования распределенных баз данных:

- прозрачность расположения данных для пользователя (иначе говоря, для пользователя распределенная база данных должна представляться и выглядеть точно так же, как и нераспределенная);

- изолированность пользователей друг от друга (пользователь должен "не чувствовать", "не видеть" работу других пользователей в тот момент, когда он изменяет, обновляет, удаляет данные);

- синхронизация и согласованность (непротиворечивость) состояния данных в любой момент времени.

Из основных вытекает ряд дополнительных принципов:

- локальная автономия (ни одна вычислительная установка для своего успешного функционирования не должна зависеть от любой другой установки);

- отсутствие центральной установки (следствие предыдущего пункта);

- независимость от местоположения (пользователю все равно, где физически находятся данные, он работает так, как будто они находятся на его локальной установке);

- непрерывность функционирования (отсутствие плановых отключений системы в целом, например для подключения новой установки или обновления версии СУБД);

- независимость от фрагментации данных (как от горизонтальной фрагментации, когда различные группы записей одной таблицы размещены на различных установках или в различных локальных базах, так и от вертикальной фрагментации, когда различные поля-столбцы одной таблицы размещены на разных установках);

- независимость от реплицирования (дублирования) данных (когда какая-либо таблица базы данных (или ее часть) физически может быть представлена несколькими копиями, расположенными на различных установках);

- распределенная обработка запросов (оптимизация запросов должна носить распределенный характер - сначала глобальная оптимизация, а далее локальная оптимизация на каждой из задействованных установок);

- распределенное управление транзакциями (в распределенной системе отдельная транзакция может требовать выполнения действий на разных установках, транзакция считается завершенной, если она успешно завершена на всех вовлеченных установках);

- независимость от аппаратуры (желательно, чтобы система могла функционировать на установках, включающих компьютеры разных типов);

- независимость от типа операционной системы (система должна функционировать вне зависимости от возможного различия ОС на различных вычислительных установках);

- независимость от коммуникационной сети (возможность функционирования в разных коммуникационных средах);

- независимость от СУБД (на разных установках могут функционировать СУБД различного типа, на практике ограничиваемые кругом СУБД, поддерживающих SQL).

В обиходе СУБД, на основе которых создаются распределенные информационные системы, также характеризуют термином "распределенные

СУБД", и, соответственно, используют термин "распределенные базы данных".

Практическая реализация распределенных вычислений осуществляется через отступление от некоторых рассмотренных выше принципов создания и функционирования распределенных систем. В зависимости от того, какой принцип приносится в "жертву" (отсутствие центральной установки, непрерывность функционирования, согласованного состояния данных и др.) выделились несколько самостоятельных направлений в технологиях распределенных систем - технологии "Клиент-сервер", технологии реплицирования, технологии объектного связывания.

Реальные распределенные информационные системы, как правило, построены на основе сочетания всех трех технологий, но в методическом плане их целесообразно рассмотреть отдельно.

5.3 Технологии и модели "Клиент-сервер"

Системы на основе технологий "Клиент-сервер" исторически выросли из первых централизованных многопользовательских автоматизированных информационных систем, интенсивно развивавшихся в 70-х годах (системы mainframe), и получили, вероятно, наиболее широкое распространение в сфере информационного обеспечения крупных предприятий и корпораций.

В технологиях "Клиент-сервер" отступают от одного из главных принципов создания и функционирования распределенных систем - отсутствия центральной установки. Поэтому можно выделить две основные идеи, лежащие в основе клиент-серверных технологий:

- общие для всех пользователей данные на одном или нескольких серверах;
- много пользователей (клиентов), на различных вычислительных установках, совместно (параллельно и одновременно) обрабатывающих общие данные.

Иначе говоря, системы, основанные на технологиях "Клиент-сервер", распределены только в отношении пользователей, поэтому часто их не относят к "настоящим" распределенным системам, а считают отдельным классом многопользовательских систем.

Важное значение в технологиях "Клиент-сервер" имеют понятия сервера и клиента.

Под сервером в широком смысле понимается любая система, процесс, компьютер, владеющие каким-либо вычислительным ресурсом (памятью, временем, производительностью процессора и т. д.).

Клиентом называется также любая система, процесс, компьютер, пользователь, запрашивающие у сервера какой-либо ресурс, пользующиеся каким-либо ресурсом или обслуживаемые сервером иным способом.

В своем развитии системы "Клиент-сервер" прошли несколько этапов, в ходе которых сформировались различные модели систем "Клиент-сервер".

Их реализация и, следовательно, правильное понимание основаны на разделении структуры СУБД на три компонента:

- компонент представления, реализующий функции ввода и отображения данных, называемый иногда еще просто как интерфейс пользователя;
- прикладной компонент, включающий набор запросов, событий, правил, процедур и других вычислительных функций, реализующий предназначение автоматизированной информационной системы в конкретной предметной области;
- компонент доступа к данным, реализующий функции хранения-извлечения, физического обновления и изменения данных.

Исходя из особенностей реализации и распределения в системе этих трех компонентов различают четыре модели технологий "Клиент-сервер":

- модель файлового сервера (File Server - FS);
- модель удаленного доступа к данным (Remote Data Access - RDA);
- модель сервера базы данных (DataBase Server - DBS);
- модель сервера приложений (Application Server - AS).

5.3.1 Модель файлового сервера

Модель файлового сервера является наиболее простой и характеризует не столько способ образования информационной системы, сколько общий способ взаимодействия компьютеров в локальной сети. Один из компьютеров сети выделяется и определяется файловым сервером, т. е. общим хранилищем любых данных. Суть FS- модели иллюстрируется схемой, приведенной на рис. 5.3.

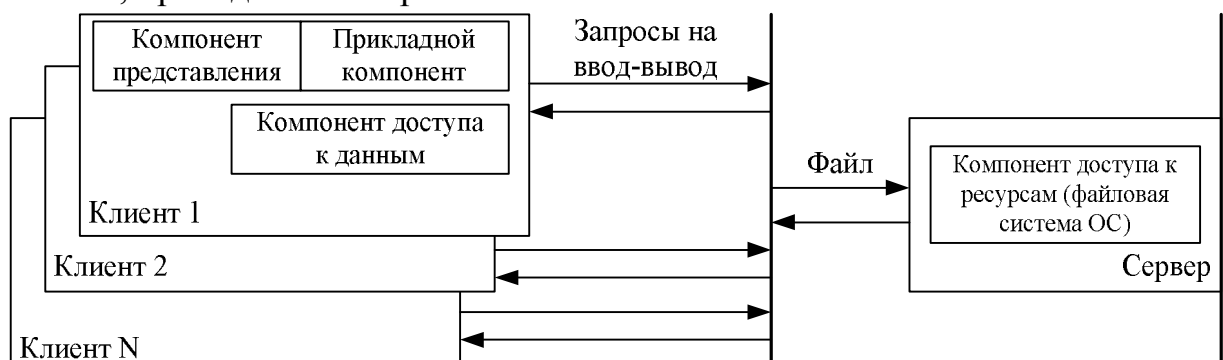


Рис 5.3 - Модель файлового сервера

В FS-модели все основные компоненты размещаются на клиентской установке. При обращении к данным ядро СУБД, в свою очередь, обращается с запросами на ввод-вывод данных за сервисом к файловой системе. С помощью функций операционной системы в оперативную память клиентской установки полностью или частично на время сеанса работы копируется файл базы данных. Таким образом, сервер в данном случае выполняет чисто пассивную функцию.

Достоинством данной модели являются ее простота, отсутствие высоких требований к производительности сервера (главное, требуемый объем дискового пространства). Следует также отметить, что программные компоненты СУБД в данном случае не распределены, т.е. никакая часть СУБД на сервере не устанавливается и не размещается.

Недостатки данной модели - высокий сетевой трафик, достигающий пиковых значений особенно в момент массового вхождения в систему пользователей, например в начале рабочего дня. Однако более существенным недостатком, с точки зрения работы с общей базой данных, является отсутствие специальных механизмов безопасности файла (файлов) базы данных со стороны СУБД. Иначе говоря, разделение данных между пользователями (параллельная работа с одним файлом данных) осуществляется только средствами файловой системы ОС для одновременной работы нескольких прикладных программ с одним файлом.

Несмотря на очевидные недостатки, модель файлового сервера является естественным средством расширения возможностей персональных (настольных) СУБД в направлении поддержки многопользовательского режима и, очевидно, в этом плане еще будет сохранять свое значение.

5.3.2 Модель удаленного доступа к данным

Модель удаленного доступа к данным основана на учете специфики размещения и физического манипулирования данными во внешней памяти для реляционных СУБД. В *RDA-модели* компонент доступа к данным в СУБД полностью отделен от двух других компонентов (компонента представления и прикладного компонента) и размещается на сервере системы.

Компонент доступа к данным реализуется в виде самостоятельной программной части СУБД, называемой SQL-сервером, и устанавливается на вычислительной установке сервера системы. Функции SQL-сервера ограничиваются низкоуровневыми операциями по организации, размещению, хранению и манипулированию данными в дисковой памяти сервера. Иначе говоря, SQL-сервер играет роль машины данных. Схема RDA-модели приведена на рис. 5.4.

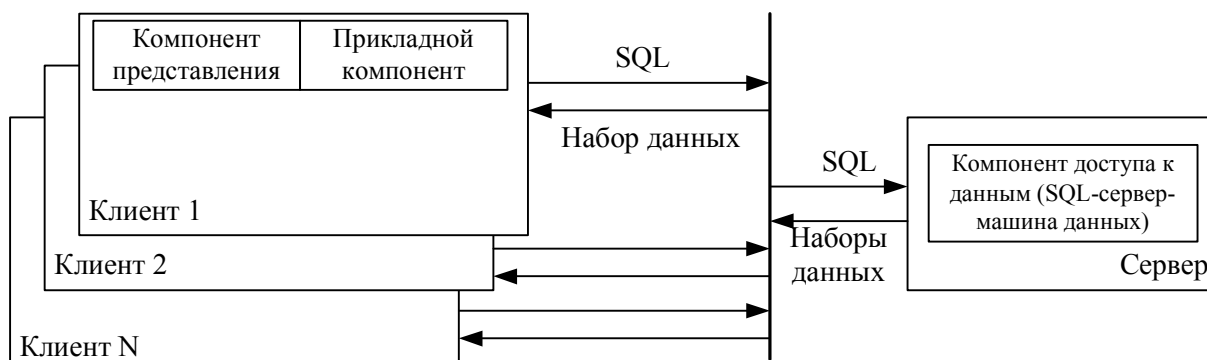


Рис 5.4. Модель удаленного доступа к данным (RDA-модель)

В файле (файлах) базы данных, размещаемом на сервере системы, находится также и системный каталог базы данных, в который помещаются в том числе и сведения о зарегистрированных клиентах, их полномочиях и т. п.

На клиентских установках инсталлируются программные части СУБД, реализующие интерфейсные и прикладные функции. Пользователь, входя в клиентскую часть системы, регистрируется через нее на сервере системы и начинает обработку данных.

Прикладной компонент системы (библиотеки запросов, процедуры обработки данных) полностью размещается и выполняется на клиентской установке. При реализации своих функций прикладной компонент формирует необходимые SQL-инструкции, направляемые SQL-серверу. SQL-сервер, представляющий специальный программный компонент, ориентированный на интерпретацию SQL-инструкций и высокоскоростное выполнение низкоуровневых операций с данными, принимает и координирует SQL-инструкции от различных клиентов, выполняет их, проверяет и обеспечивает выполнение ограничений целостности данных и направляет клиентам результаты обработки SQL-инструкций, представляющие, как известно, наборы (таблицы) данных.

Таким образом, общение клиента с сервером происходит через SQL-инструкции, а с сервера на клиентские установки передаются только результаты обработки, т. е. наборы данных, которые могут быть существенно меньше по объему всей базы данных. В результате резко уменьшается загрузка сети, а сервер приобретает активную центральную функцию. Кроме того, ядро СУБД в виде SQL-сервера обеспечивает также традиционные и важные функции по обеспечению ограничений целостности и безопасности данных при совместной работе нескольких пользователей.

Другим, может быть неявным, достоинством RDA-модели является унификация интерфейса взаимодействия прикладных компонентов информационных систем с общими данными. Такое взаимодействие стандартизовано в рамках языка SQL специальным протоколом ODBC (Open Database Connectivity - открытый доступ к базам данных), играющим важную роль в обеспечении интероперабельности (многопротокольность), т.е. независимости от типа СУБД на клиентских установках в распределенных системах.

Интероперабельность (многопротокольность) СУБД - способность СУБД обслуживать прикладные программы, первоначально ориентированные на разные типы СУБД. Иначе говоря, специальный компонент ядра СУБД на сервере (так называемый драйвер ODBC) способен воспринимать, обрабатывать запросы и направлять результаты их обработки на клиентские установки, функционирующие под управлением реляционных СУБД других, не "родных" типов.

Такая возможность существенно повышает гибкость в создании распределенных информационных систем на базе интеграции уже существующих в какой-либо организации локальных баз данных под управлением настольных или другого типа реляционных СУБД.

К недостаткам RDA-модели можно отнести высокие требования к клиентским вычислительным установкам, так как прикладные программы обработки данных, определяемые спецификой предметной области информационной системы, выполняются на них.

Другим недостатком является все же существенный трафик сети, обусловленный тем, что с сервера базы данных клиентам направляются наборы (таблицы) данных, которые в определенных случаях могут занимать достаточно существенный объем.

5.3.3 Модель сервера базы данных

Развитием PDA-модели стала модель сервера базы данных. Ее сердцевиной является механизм хранимых процедур. В отличие от PDA-модели, определенные для конкретной предметной области информационной системы события, правила и процедуры, описанные средствами языка SQL, хранятся вместе с данными на сервере системы и на нем же выполняются. Иначе говоря, прикладной компонент полностью размещается и выполняется на сервере системы. Схематично DBS-модель приведена на рис. 2.5.

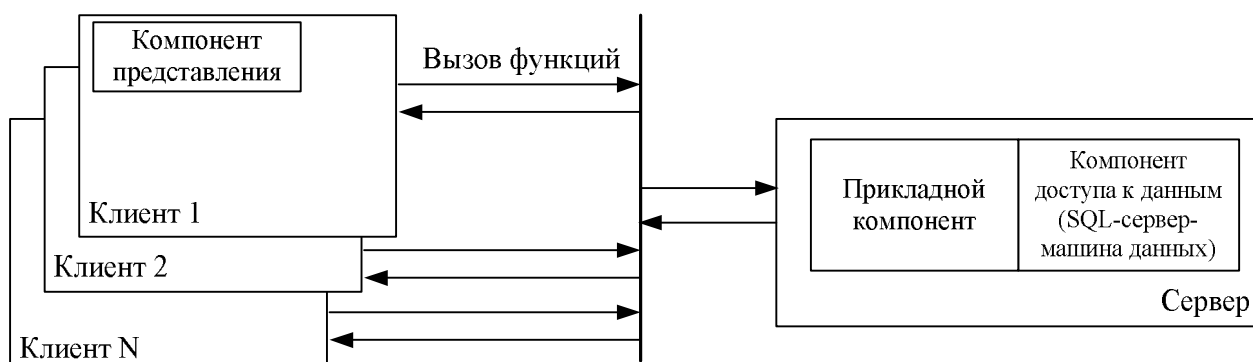


Рис. 5.5 Модель сервера базы данных (DBS-модель)

На клиентских установках в DBS-модели размещается только интерфейсный компонент (компонент представления), что существенно снижает требования к вычислительной установке клиента. Пользователь через интерфейс системы на клиентской установке направляет на сервер базы данных только лишь вызовы необходимых процедур, запросов и других функций по обработке данных. Все затратные операции по доступу и обработке данных выполняются на сервере и клиенту направляются лишь результаты обработки, а не наборы данных, как в RDA-модели. Этим обеспечивается существенное снижение трафика сети в DBS-модели по сравнению с RDA -моделью.

Следует заметить, что на сервере системы выполняются процедуры прикладных задач одновременно всех пользователей системы. В результате резко возрастают требования к вычислительной установке сервера, причем

как к объему дискового пространства и оперативной памяти, так и к быстродействию. Это основной *недостаток* DBS-модели.

К достоинствам же DBS-модели, помимо разгрузки сети, относится и более активная роль сервера сети, размещение, хранение и выполнение на нем механизма событий, правил и процедур, возможность более адекватно и эффективно "настраивать" распределенную информационную систему на все нюансы предметной области.

Также более надежно обеспечивается согласованность состояния и изменения данных и, вследствие этого, повышается надежность хранения и обработки данных, эффективно координируется коллективная работа пользователей с общими данными.

5.3.4 Модель сервера приложений

Чтобы разнести требования к вычислительным ресурсам сервера в отношении быстродействия и памяти по разным вычислительным установкам, используется модель сервера приложений.

Суть AS-модели заключается в переносе прикладного компонента информационной системы на специализированный в отношении повышенных ресурсов по быстродействию дополнительный сервер системы. Схема AS-модели приведена на рис. 5.6

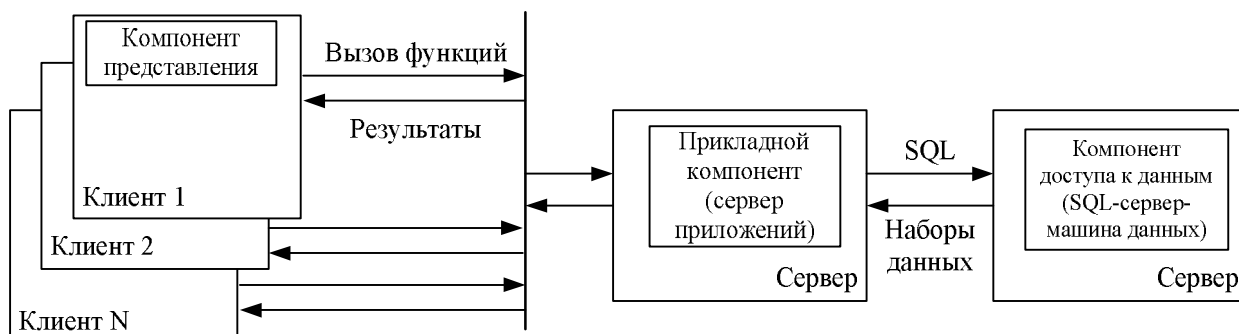


Рис. 5.6. Модель сервера приложений (AS-модель)

Как и в DBS-модели, на клиентских установках располагается только интерфейсная часть системы, т. е. компонент представления. Однако вызовы функций обработки данных направляются на сервер приложений, где эти функции совместно выполняются для всех пользователей системы. За выполнением низкоуровневых операций по доступу и изменению данных сервер приложений, как в RDA-модели, обращается к SQL-серверу, направляя ему вызовы SQL-процедур, и получая, соответственно, от него наборы данных.

Как известно, последовательная совокупность операций над данными (SQL-инструкций), имеющая отдельное смысловое значение, называется *транзакцией*.

В этом отношении сервер приложений управляет формированием транзакций, которые выполняет SQL-сервер. Поэтому программный компонент СУБД, устанавливаемый на сервере приложений, еще называют монитором обработки транзакций (Transaction Processing Monitors - TRM), или просто *монитором транзакций*.

AS-модель, сохраняя сильные стороны DBS-модели, позволяет оптимально построить вычислительную схему информационной системы, однако, как и в случае RDA-модели, повышает трафик сети.

В практических случаях используются смешанные модели, когда простейшие прикладные функции и обеспечение ограничений целостности данных поддерживаются хранимыми на сервере процедурами (DBS-модель), а более сложные функции предметной области (так называемые правила бизнеса) реализуются прикладными программами на клиентских установках (RDA-модель) или на сервере приложений (AS-модель).

5.4 Технологии объектного связывания данных

Унификация взаимодействия прикладных компонентов с ядром информационных систем в виде SQL-серверов, наработанная для клиент-серверных систем, позволила выработать аналогичные решения и для интеграции разрозненных локальных баз данных под управлением настольных СУБД в сложные децентрализованные гетерогенные распределенные системы. Такой подход получил название *объектного связывания данных*.

С узкой точки зрения, технология объектного связывания данных решает задачу обеспечения доступа из одной локальной базы, открытой одним пользователем, к данным в другой локальной базе (в другом файле), возможно находящейся на другой вычислительной установке, открытой и эксплуатируемой другим пользователем.

Решение этой задачи основывается на поддержке современными "настольными" СУБД (MS Access, MS FoxPro, dBase и др.) технологии "объектов доступа к данным" - DAO.

При этом следует отметить, что под объектом понимается интеграция данных и методов, их обработки в одно целое (объект), на чем основываются объектно-ориентированное программирование и современные объектно-ориентированные операционные среды. Другими словами, СУБД, поддерживающие DAO, получают возможность внедрять и оперировать в локальных базах объектами доступа к данным, физически находящимся в других файлах, возможно на других вычислительных установках и под управлением других СУБД.

Технически технология DAO основана на уже упоминавшемся протоколе ODBC, который принят за стандарт доступа не только к данным на SQL-серверах клиент-серверных систем, но и в качестве стандарта доступа к любым данным под управлением реляционных СУБД.

Непосредственно для доступа к данным на основе протокола ODBC используются специальные программные компоненты, называемые драйверами ODBC (инициализируемые на тех установках, где находятся данные).

Схематично принцип и особенности доступа к внешним базам данных на основе объектного связывания иллюстрируются на рис. 5.7.

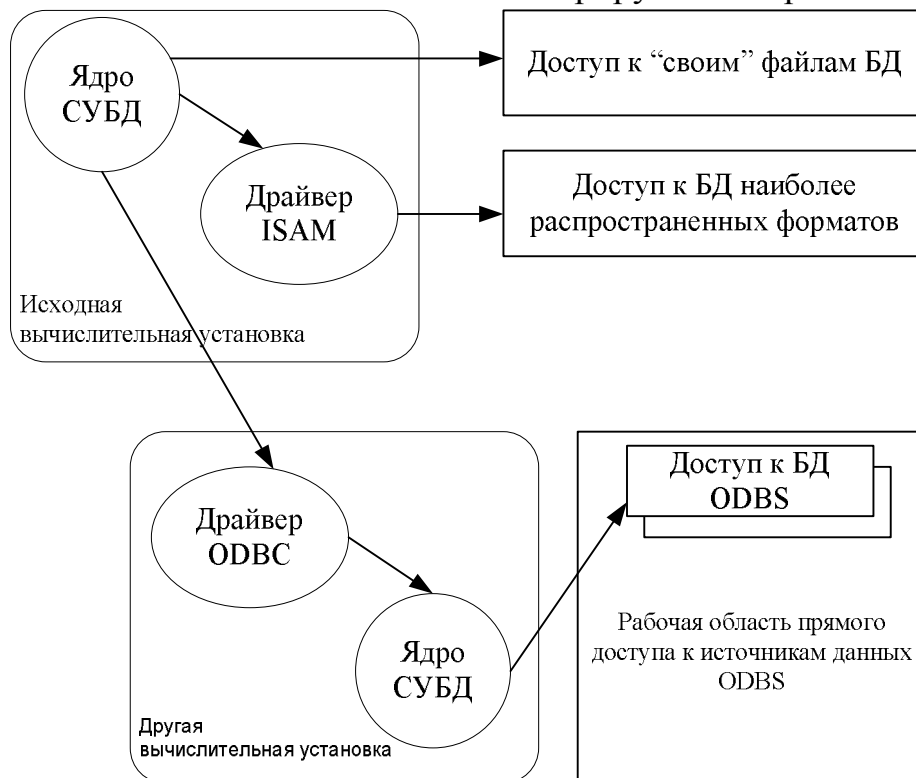


Рис. 5.7 - Принцип доступа к внешним данным па основе ODBC

Прежде всего, современные настольные СУБД обеспечивают возможность прямого доступа к объектам (таблицам, запросам, формам) внешних баз данных "своих" форматов. Иначе говоря, в открытую в текущем сеансе работы базу данных пользователь имеет возможность вставить специальные ссылки-объекты и оперировать с данными из другой (внешней, т. е. не открываемой специально в данном сеансе) базы данных.

Объекты из внешней базы данных, вставленные в текущую базу данных, называются *связанными* и, как правило, имеют специальные обозначения для отличия от внутренних объектов. При этом следует подчеркнуть, что сами данные физически в файл (файлы) текущей базы данных не помещаются, а остаются в файлах своих баз данных. В системный каталог текущей базы данных помещаются все необходимые для доступа сведения о связанных объектах - внутреннее имя и внешнее, т. е. истинное имя объекта во внешней базе данных, полный путь к файлу внешней базы и г. п.

Связанные объекты для пользователя ничем не отличаются от внутренних объектов. Пользователь может также открывать связанные во внешних базах таблицы данных, осуществлять поиск, изменение, удаление и

добавление данных, строить запросы по таким таблицам и т. д. Связанные объекты можно интегрировать в схему внутренней базы данных, т. е. устанавливать связи между внутренними и связанными таблицами.

Технически оперирование связанными объектами из внешних баз данных "своего" формата мало отличается от оперирования с данными из текущей базы данных.

Ядро СУБД при обращении к данным связанного объекта по системному каталогу текущей базы данных находит сведения о месте нахождения и других параметрах соответствующего файла (файлов) внешней базы данных и прозрачно (т. е. невидимо для пользователя) открывает этот файл (файлы). Далее обычным порядком организует в оперативной памяти буферизацию страниц внешнего файла данных для непосредственно доступа и манипулирования данными.

Следует также заметить, что на основе возможностей многопользовательского режима работы с файлами данных современных операционных систем, с файлом внешней базы данных, если он находится на другой вычислительной установке, может в тот же момент времени работать и другой пользователь, что и обеспечивает коллективную обработку общих распределенных данных.

Подобный принцип построения распределенных систем при больших объемах данных в связанных таблицах приведет к существенному увеличению трафика сети, так как по сети постоянно передаются даже не наборы данных, а страницы файлов баз данных, что может приводить к пиковым перегрузкам сети. Поэтому представленные схемы локальных баз данных со взаимными связанными объектами нуждаются в дальнейшей тщательной проработке.

Не менее существенной проблемой является отсутствие надежных механизмов безопасности данных и обеспечения ограничений целостности. Совместная работа нескольких пользователей с одними и теми же данными обеспечивается только функциями операционной системы по одновременному доступу к файлу нескольких приложений.

Аналогичным образом обеспечивается доступ к данным, находящимся в базах данных наиболее распространенных форматов других СУБД, таких, например, как базы данных СУБД FoxPro, dBASE.

При этом доступ может обеспечиваться как непосредственно ядром СУБД, так и специальными дополнительными драйверами ISAM (Indexed Sequential Access Method), входящими, как правило, в состав комплекта СУБД.

Объектное связывание ограничивается только непосредственно таблицами данных, исключая другие объекты базы данных (запросы, формы, отчеты), реализация и поддержка которых зависят от специфики конкретной СУБД.

Определенной проблемой технологий объектного связывания является появление "брешей" в системах защиты данных и разграничения доступа. Вызовы драйверов ODBC для осуществления процедур доступа к данным

помимо пути, имени файлов и требуемых объектов (таблиц), если соответствующие базы защищены, содержат в открытом виде пароли доступа, в результате чего может быть проанализирована и раскрыта система разграничения доступа и защиты данных.

5.5 Технологии реплицирования данных

Во многих случаях узким местом распределенных систем, построенных на основе технологий "Клиент-сервер" или объектного связывания данных, является недостаточно высокая производительность из-за необходимости передачи по сети большого количества данных. Определенную альтернативу построения быстродействующих распределенных систем предоставляют технологии реплицирования данных.

Реплика называют особую копию базы данных для размещения на другом компьютере сети с целью автономной работы пользователей с одинаковыми (согласованными) данными общего пользования.

Основная идея реплицирования заключается в том, что пользователи работают автономно с одинаковыми (общими) данными, растиражированными по локальным базам данных, обеспечивая с учетом отсутствия необходимости передачи и обмена данными по сети максимальную для своих вычислительных установок производительность.

Тиражирование (или репликация,) - создание дублирующих копий (репликатов) объектов данных на разных узлах с целью повышения доступности и/или сокращения времени доступа к критически важным данным.

Программное обеспечение СУБД для реализации такого подхода соответственно дополняется функциями тиражирования (реплицирования) баз данных, включая тиражирование как самих данных и их структуры, так и системного каталога с информацией о размещении реплик, иначе говоря, с информацией о конфигурировании построенной таким образом распределенной системы.

При этом, однако, возникают две проблемы обеспечения одного из основополагающих принципов построения и функционирования распределенных систем (а именно, - непрерывности согласованного состояния данных):

- обеспечение согласованного состояния во всех репликах количества и значений общих данных;
- обеспечение согласованного состояния во всех репликах структуры данных.

Обеспечение согласованного состояния общих данных, в свою очередь, основывается на реализации одного из двух принципов:

- принципа непрерывного размножения обновлений (любое обновление данных в любой реплике должно быть немедленно размножено);
- принципа отложенных обновлений (обновления реплик могут быть отложены до специальной команды или ситуации).

Принцип непрерывного размножения обновлений является основополагающим при построении так называемых систем реального времени, таких, например, как системы управления воздушным движением, системы бронирования билетов пассажирского транспорта и т.п., где требуется непрерывное и точное соответствие реплик или других растиражированных данных во всех узлах и компонентах подобных распределенных систем.

Реализация принципа непрерывного размножения обновлений заключается в том, что любая транзакция считается успешно завершенной, если она успешно завершена на всех репликах системы. На практике реализация этого принципа встречает существенные затруднения.

В целом ряде предметных областей распределенных информационных систем режим реального времени с точки зрения непрерывности согласования данных не требуется. Такие системы автоматизируют те организационно-технологические структуры, в которых информационные процессы не столь динамичны. В этом случае обновление реплик распределенной информационной системы, если она будет построена на технологии реплицирования, требуется, скажем, только лишь один раз за каждый рабочий час, или за каждый рабочий день.

Такого рода информационные системы строятся на основе принципа отложенных обновлений. Накопленные в какой-либо реплике изменения данных специальной командой пользователя направляются для обновления всех остальных реплик систем. Такая операция называется ***синхронизацией реплик***.

Решение второй проблемы согласованности данных, а именно - согласованности структуры данных, осуществляется через частичное отступление, как и в системах "Клиент-сервер", от принципа отсутствия центральной установки и основывается на технике ***главной реплики***, т.е. одна из реплик базы данных объявляется главной. При этом изменять структуру базы данных можно только в главной реплике. Эти изменения структуры данных тиражируются на основе принципа отложенных обновлений, т.е. через специальную синхронизацию реплик.

Частичность отступления от принципа отсутствия центральной установки заключается в том, что в отличие от чисто централизованных систем, выход из строя главной реплики не влечет сразу гибель всей распределенной системы, так как остальные реплики продолжают функционировать автономно. Более того, на практике СУБД, поддерживающие технологию реплицирования, позволяют пользователю с определенными полномочиями (администратору системы) преобразовать любую реплику в главную и тем самым полностью восстановить работоспособность всей системы.

Технологии репликации данных в тех случаях, когда не требуется обеспечивать большие потоки и интенсивность обновляемых в информационной сети данных, являются экономичным решением проблемы создания распределенных информационных систем с элементами

централизации по сравнению с использованием дорогостоящих клиент-серверных систем.

На практике для совместной коллективной обработки данных применяются смешанные технологии, включающие элементы объектного связывания данных, репликаций и клиент-серверных решений. При этом дополнительно к проблеме логического проектирования, т. е. проектирования логической схемы организации данных (таблицы, поля, ключи, связи, ограничения целостности), добавляется не менее сложная проблема транспортно-технологического проектирования информационных потоков, разграничения доступа и т. д. К сожалению, пока не проработаны теоретико-методологические и инструментальные подходы для автоматизации проектирования распределенных информационных систем с учетом факторов как логики, так и информационно-технологической инфраструктуры предметной области.

Тем не менее, развитие и все более широкое распространение распределенных информационных систем, определяемое самой распределенной природой информационных потоков и технологий, является основной перспективой развития автоматизированных информационных систем.

6.1 Понятие о компьютерном математическом моделировании.

6.1.1 Общие сведения о компьютерном математическом моделировании

Модель - материальный объект, система математических зависимостей или программа, имитирующая структуру или функционирование исследуемого объекта.

Моделирование - представление различных характеристик поведения физической или абстрактной системы с помощью другой системы.

Математическое моделирование - метод исследования процессов и явлений на их математических моделях.

Изучение компьютерного математического моделирования открывает широкие возможности для осознания связи информатики с математикой и другими науками - естественными и социальными. Компьютерное математическое моделирование в разных своих проявлениях использует практически весь аппарат современной математики.

Математическое моделирование не всегда требует компьютерной поддержки. Каждый специалист, профессионально занимающийся математическим моделированием, делает все возможное для аналитического исследования модели. Аналитические решения (т.е. представленные формулами, выражающими результаты исследования через исходные данные) обычно удобнее и информативнее численных. Возможности аналитических методов решения сложных математических задач, однако, очень ограничены и, как правило, эти методы гораздо сложнее численных. В компьютерном моделировании доминируют численные методы, реализуемые на компьютерах. Однако понятия "аналитическое решение" и "компьютерное решение" отнюдь не противостоят друг другу, так как:

а) все чаще компьютеры при математическом моделировании используются не только для численных расчетов, но и для аналитических преобразований:

б) результат аналитического исследования математической модели часто выражен столь сложной формулой, что при взгляде на нее не складывается восприятия описываемого ей процесса. Эту формулу нужно представить графически, проиллюстрировать в динамике, иногда даже озвучить, т.е. проделать то, что называется "визуализацией абстракций". При этом компьютер - незаменимое техническое средство.

6.1.2 Классификация математических моделей

К классификации математических моделей можно подходить по-разному, положив в основу классификации различные принципы.

1) Классификация моделей по отраслям наук (математические модели в физике, биологии, социологии и т.д.);

2) Классификация моделей по применяемому математическому аппарату (модели, основанные на применении обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, стохастических методов, дискретных алгебраических преобразований и т.д.);

3) Классификация моделей с точки зрения целей моделирования.

- дескриптивные (описательные) модели;
- оптимизационные модели;
- многокритериальные модели;
- игровые модели;
- имитационные модели.

Пример.

1) Моделируя движение кометы, вторгшейся в Солнечную систему, мы описываем (предсказываем) траекторию ее полета, расстояние, на котором она пройдет от Земли и т.д., т.е. ставим чисто описательные цели. У нас нет никаких возможностей повлиять на движение кометы, что-то изменить.

2) Меняя тепловой режим в зернохранилище, мы можем стремиться подобрать такой, чтобы достичь максимальной сохранности зерна, т.е. оптимизируем процесс.

Часто приходится оптимизировать процесс по нескольким параметрам сразу, причем цели могут быть весьма противоречивыми. Например, зная цены на продукты и потребность человека в пище, организовать питание больших групп людей (в армии, летнем лагере и др.) как можно полезнее и как можно дешевле.

3) Игровые модели могут иметь отношение не только к детским играм (в том числе и компьютерным), но и к вещам весьма серьезным.

4) Бывает, что модель в большой мере подражает реальному процессу, т.е. имитирует его.

Имитационная модель - описание системы и ее поведения, которое может быть реализовано и исследовано в ходе операций на компьютере.

Имитационное моделирование - исследование поведения сложной системы на ее модели.

Можно сказать, что чаще всего имитационное моделирование применяется для того, чтобы описать свойства большой системы при условии, что поведение составляющих ее объектов очень просто и четко сформулировано. Математическое описание тогда сводится к уровню статистической обработки результатов моделирования при нахождении макроскопических характеристик системы. Такой компьютерный эксперимент фактически претендует на воспроизведение натурального эксперимента.

Имитационное моделирование позволяет осуществить проверку гипотез, исследовать влияние различных факторов и параметров.

6.2 Этапы, цели и средства компьютерного математического моделирования

Здесь мы рассмотрим процесс компьютерного математического моделирования, включающий численный эксперимент с моделью (рис. 6.1).

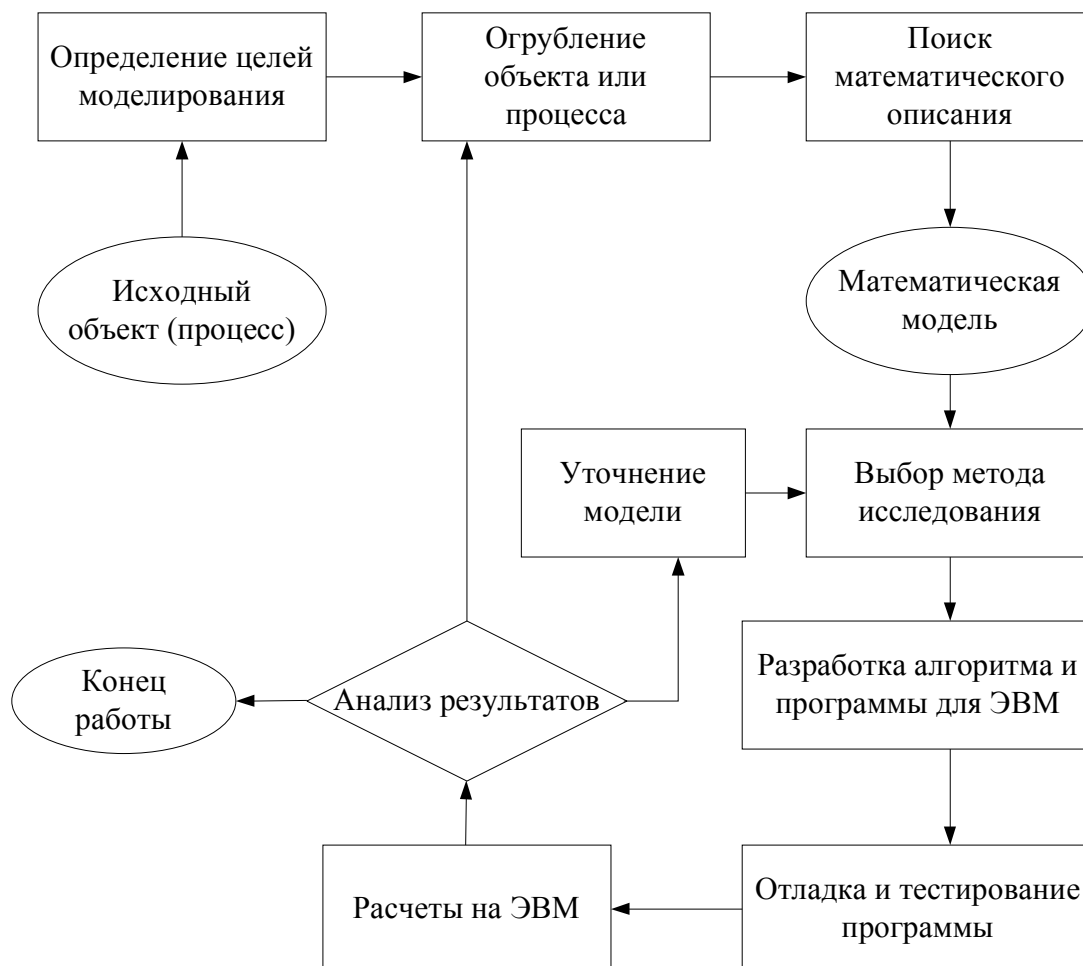


Рис. 6.1 - Общая схема процесса компьютерного математического моделирования

Первый этап - определение целей моделирования.

Основные из них таковы:

- 1) модель нужна для того, чтобы понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействия с окружающим миром (понимание);
- 2) модель нужна для того, чтобы научиться управлять объектом (или процессом) и определить наилучшие способы управления при заданных целях и критериях (управление);
- 3) модель нужна для того, чтобы прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект (прогнозирование).

Выработка концепции управления объектом - другая возможная цель моделирования. Какой режим полета самолета выбрать для того, чтобы полет был вполне безопасным и экономически наиболее выгодным? Как составить график выполнения сотен видов работ на строительстве большого объекта, чтобы оно закончилось в максимально короткий срок? Множество таких проблем систематически возникает перед экономистами, конструкторами, учеными.

Наконец, *прогнозирование последствий* тех или иных воздействий на объект может быть как относительно простым делом в несложных физических системах, так и чрезвычайно сложным - на грани выполнимости - в системах биолого-экономических, социальных. Если относительно легко ответить на вопрос об изменении режима распространения тепла в тонком стержне при изменениях в составляющем его сплаве, то несравненно труднее проследить (предсказать) экологические и климатические последствия строительства крупной ГЭС или социальные последствия изменений налогового законодательства. Возможно, и здесь методы математического моделирования будут оказывать в будущем более значительную помощь.

Составим список величин, от которых зависит поведение объекта или ход процесса, а также тех величин, которые желательно получить в результате моделирования. Обозначим первые (входные) величины через x_1, x_2, \dots, x_n ; вторые (выходные) через y_1, y_2, \dots, y_k .

Символически поведение объекта или процесса можно представить в виде: $y_j = F_j(x_1, x_2, \dots, x_n) \quad (j = 1, 2, \dots, k)$,

где F - те действия, которые следует произвести над входными параметрами, чтобы получить результаты.

Входные параметры, могут быть известны "точно", т.е. поддаваться (в принципе) измерению однозначно и с любой степенью точности - тогда они являются детерминированными величинами. Так, в классической механике, сколь сложной ни была бы моделируемая система, входные параметры детерминированы - соответственно, детерминирован, однозначно развивается во времени процесс эволюции такой системы.

Однако в природе и обществе гораздо чаще встречаются процессы иного рода, когда значения входных параметров известны лишь с определенной степенью вероятности, т.е. эти параметры, являются вероятностными (стохастическими), и, соответственно, таким же является процесс эволюции системы (случайный процесс).

"Случайный" - не значит "непредсказуемый"; просто характер исследования, задаваемых вопросов резко меняется (они приобретают вид "С какой вероятностью...", "С каким математическим ожиданием..." и т.п.). Примеров случайных процессов не счесть как в науке, так и в обыденной жизни (силы, действующие на летящий самолет в ветреную погоду, переход улицы при большом потоке транспорта и т.д.).

Для стохастической модели выходные параметры могут быть как величинами вероятностными, так и однозначно определяемыми.

Важнейшим этапом моделирования является разделение входных параметров по степени важности влияния их изменений на выходные. Такой процесс называется **ранжированием** (разделением по рангам). Чаще всего невозможно (да и не нужно) учитывать все факторы, которые могут повлиять на значения интересующих нас величин y .

От того, насколько умело выделены важнейшие факторы, зависит успех моделирования, быстрота и эффективность достижения цели. Выделить более важные (или, как говорят, значимые) факторы и отсеять менее важные может лишь специалист в той предметной области, к которой относится модель.

Отбрасывание (по крайней мере при первом подходе) менее значимых факторов огрубляет объект моделирования и способствует пониманию его главных свойств и закономерностей. Умело ранжированная модель должна быть адекватна исходному объекту или процессу в отношении целей моделирования. Обычно определить, адекватна ли модель, можно только в процессе экспериментов с ней, анализа результатов.

Следующий этап - поиск математического описания. На этом этапе необходимо перейти от абстрактной формулировки модели к формулировке, имеющей конкретное математическое наполнение. В этот момент модель предстает перед нами в виде уравнения, системы уравнений, системы неравенств, дифференциального уравнения или системы таких уравнений и т.д.

Когда математическая модель сформулирована, выбирается **метод ее исследования**. Как правило, для решения одной и той же задачи есть несколько конкретных методов, различающихся эффективностью, устойчивостью и т.д. От верного выбора метода часто зависит успех всего процесса.

Разработка алгоритма и составление программы для ЭВМ - это творческий и трудноформализуемый процесс. В настоящее время при компьютерном математическом моделировании часто используются приемы процедурно-ориентированного (структурного) программирования.

При создании имитационной модели можно также воспользоваться возможностями одного из пакетов математической поддержки (MATHEMATICA, MathCad, MathLab и др).

В настоящее время существуют проблемно-ориентированные имитационные языки, в которых объединяются различные альтернативные подходы, и которые самой своей структурой определяют возможную схему действий разработчика модели. Характерным примером такого рода является имитационный язык СЛАМ II (SLAM - Simulating Language for Alternative Modeling имитационный язык для альтернативного моделирования).

После составления программы решаем с ее помощью простейшую **тестовую задачу** (желательно, с заранее известным ответом) с целью устранения грубых ошибок. Это - лишь начало процедуры тестирования, которую трудно описать формально исчерпывающим образом. По существу, тестирование может продолжаться долго и закончиться тогда, когда

пользователь по своим профессиональным признакам сочтет программу верной.

Затем следует собственно численный эксперимент, и выясняется, соответствует ли модель реальному объекту (процессу). Модель адекватна реальному процессу, если некоторые характеристики процесса, полученные на ЭВМ, совпадают с экспериментальными с заданной степенью точности. В случае несоответствия модели реальному процессу возвращаемся к одному из предыдущих этапов.

6.2.1 Моделирования случайных процессов

Моделирование случайных процессов - мощнейшее направление в современном математическом моделировании.

Событие называется случайным, если оно достоверно непредсказуемо. Случайность окружает наш мир и чаще всего играет отрицательную роль в нашей жизни. Однако есть обстоятельства, в которых случайность может оказаться полезной. В сложных вычислениях, когда искомый результат зависит от результатов многих факторов, моделей и измерений, можно сократить объем вычислений за счет случайных значений значащих цифр.

При вероятностном моделировании используют различные методы, которые позволяют решать задачи из различных областей. Ниже перечислены сферы применения вероятностных методов.

Метод статистического моделирования: решение краевых задач математической физики, решение систем линейных алгебраических уравнений, обращение матриц и сводящиеся к ним сеточные методы решения систем дифференциальных уравнений, вычисление кратных интегралов, решение интегральных уравнений, задач ядерной физики, газовой динамики, фильтрации, теплотехники.

Метод имитационного моделирования: моделирование систем массового обслуживания, задачи АСУ, АСУП и АСУТП, задачи защиты информации, моделирование сложных игровых ситуаций и динамических систем.

Метод стохастической аппроксимации: рекуррентные алгоритмы решения задач статистического оценивания.

Метод случайного поиска: решение задач оптимизации систем, зависящих от большого числа параметров, нахождение экстремумов функции большого числа переменных.

Другие методы: вероятностные методы распознавания образов, модели адаптации, обучения и самообучения.

При компьютерном математическом моделировании случайных процессов нельзя обойтись без наборов так называемых случайных чисел, удовлетворяющих заданному закону распределения. На самом деле эти числа генерирует компьютер по определенному алгоритму, т.е. они не являются вполне случайными хотя бы потому, что при повторном запуске программы с

теми же параметрами последовательность повторится; такие числа называют "псевдослучайными".

Для не слишком требовательного пользователя обычно достаточно возможности датчика (генератора) случайных чисел, встроенного в большинство языков программирования. Так, в языке Паскаль есть функция `random`, значения которой - случайные числа из диапазона $[0, 1]$. Ее использованию обычно предшествует использование процедуры `randomize`, служащей для начальной "настройки" датчика, т.е. получения при каждом из обращений к датчику разных последовательностей случайных чисел. Для задач, Решение которых требует очень длинных некоррелированных последовательностей, вопрос усложняется и требует нестандартных

6.2.2 Особенности имитационного моделирования производственных систем

Для анализа производственных систем, которые очень сложны, разноплановы, не имеют исчерпывающего математического описания, а также проходят ряд этапов проектирования, реализации и развития, адекватные математические модели, будь то логические или числовые, построить не представляется возможным. Естественным здесь является использование методов имитационного моделирования.

Система может быть однозначно описана набором значений производственных параметров, характерных для каждого конкретного ее состояния. Если эти значения внести в компьютер, то изменения их в ходе вычислительного процесса можно интерпретировать как имитацию перехода системы из одного состояния в другое. При таких предположениях имитационное моделирование можно рассматривать как динамическое представление системы путем продвижения ее одного состояния к другому по характерным для нее операционным правилам.

При имитационном моделировании производственных систем изменения их состояния происходят в дискретные моменты времени. Основная концепция имитационного моделирования системы и в этом случае состоит в отображении изменений ее состояния с течением времени. Таким образом, здесь определяющим является выделение и однозначное описание состояний моделируемой системы.

Имитационные модели позволяют без использования каких-либо аналитических или других функциональных зависимостей отображать сложные объекты, состоящие из разнородных элементов, между которыми существуют разнообразные связи. В эти модели может быть включен также и человек.

Без принципиальных усложнений в такие модели могут быть включены как детерминированные, так и стохастические потоки (материальные и информационные). С помощью имитационного моделирования можно отображать взаимосвязи между рабочими местами, потоками материалов и изделий, транспортными средствами и персоналом.

Несмотря на такие очевидные преимущества, прежде всего заключающиеся в широте и универсальности применения, при этом методе из вида упускается существование логических связей, что исключает возможность полной оптимизации получаемых на этой модели решений. Гарантируется лишь возможность отбора лучшего из просмотренных вариантов.

Практически же имитационное моделирование во многих реальных случаях - единственно возможный способ исследования. После разработки имитационной модели с ней проводятся компьютерные эксперименты, которые позволяют сделать выводы о поведении производственной системы.

Появление и развитие методов компьютерного имитационного моделирования стало возможным также и в результате развития метода статистических испытаний, позволившего моделировать случайные события и процессы, занимающие большое место в реальных производствах.

При составлении имитационной модели и проведении с ее помощью моделирования исследуемого объекта необходимо решение нескольких связанных между собой задач. К ним относятся:

- анализ моделируемой системы и составление ее формализованного описания, включая выявление информационно-логической структуры системы, идентификацию ее компонентов, выбор параметров, характеризующих состояние этих компонентов, разработку компьютерной модели системы, способной воспроизвести ее поведение, планирование эксперимента по развертыванию событий в компьютерной модели, отображающих события в моделируемой системе;
- разработка методологии компьютерного статистического эксперимента, включая генерацию случайных или псевдослучайных чисел, имитацию различных случайных событий, статистическую обработку данных;
- проведение собственно компьютерного эксперимента на имитационной модели, включая управление параметрами и переменными модели в ходе ее исследования на компьютере.

Раздел 7. ТЕХНОЛОГИИ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

7.1 Общая характеристика технологии создания программного обеспечения

К программно-инструментальным средствам в первую очередь относятся алгоритмические языки и соответствующие им трансляторы, затем системы управления базами данных (СУБД) с языковыми средствами программирования в их среде, электронные таблицы с соответствующими средствами их настройки и т.п.

Рассмотрим этапы разработки программ.

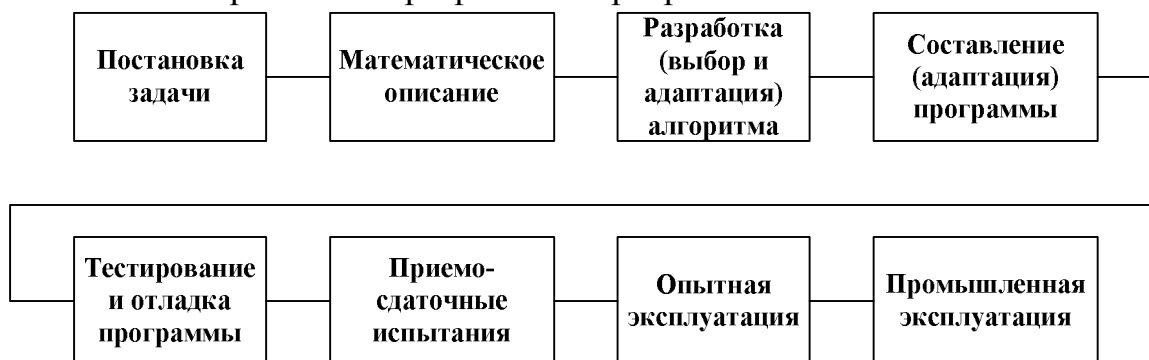


Рис.7.1 - Этапы разработки программного обеспечения.

Первый этап представляет собой **постановку задачи**. На этом этапе раскрывается сущность задачи, т.е. формулируется цель ее решения; определяется взаимосвязь с другими задачами; указывается периодичность решения; устанавливаются состав и формы представления входной, промежуточной и результатной информации.

Особое внимание в процессе постановки задачи уделяется детальному описанию входной, выходной (результатной) и промежуточной информации.

Особенность реализации этого этапа технологического процесса заключается в том, что конечный пользователь разрабатываемой программы, хорошо знающий ее проблемную сторону, обычно хуже представляет специфику и возможности использования ЭВМ для решения задачи. В свою очередь, предметная область пользователя (особенно ее отдельные нюансы, способные оказать влияние на решение задачи) зачастую незнакома разработчику программы, хотя он знает возможности и ограничения на применение ЭВМ. Именно эти противоречия являются основной причиной возникновения ошибок при реализации данного этапа технологического процесса разработки программ, которые затем неизбежно отражаются и на последующих этапах. Отсюда вся важность и ответственность этого этапа, требующего осуществления корректной и полной постановки задачи, а также необходимости однозначного ее понимания как разработчиком программы, так и ее пользователем.

Второй этап в технологии разработки программ - **математическое описание задачи** и выбор метода ее решения. Наличие этого этапа обуславливается рядом причин, одна из которых вытекает из свойства неоднозначности естественного языка, на котором описывается постановка задачи. В связи с этим на нем выполняется формализованное описание задачи, т.е. устанавливаются и формулируются средствами языка математики логико-математические зависимости между исходными и результатными данными. Математическое описание задачи обеспечивает ее однозначное понимание пользователем и разработчиком программы.

Сложность и ответственность этапа математического описания задачи и выбора (разработки) соответствующего метода ее решения часто требуют привлечения квалифицированных специалистов области прикладной математики, обладающих знанием таких дисциплин, как исследование операций, математическая статистика, вычислительная математика и т.п.

Третий этап технологического процесса подготовки решения задач ЭВМ представляет собой **алгоритмизацию** ее решения, т.е. разработку оригинального или адаптацию (уточнение и корректировку) уже известного алгоритма.

Алгоритмизация - это сложный творческий процесс. В основу процесса алгоритмизации положено фундаментальное понятие математики и программирования - алгоритм. Алгоритм - это конечный набор правил, однозначно раскрывающих содержание и последовательность выполнения операций для систематического решения определенного класса задач за конечное число шагов.

Любой алгоритм обладает следующими важными свойствами: Детерминированностью, массовостью, результатностью и дискретностью.

Детерминированность алгоритма (определенность, однозначность) - свойство, определяющее однозначность результата работы алгоритма при одних и тех же исходных данных. Это означает, что набор указаний алгоритма должен быть однозначно и точно понят любым исполнителем.

Массовость алгоритма - свойство, определяющее пригодность использования алгоритма для решения множества задач данного класса. Оно предполагает возможность варьирования исходными данными в определенных пределах. Свойство массовости алгоритма является определяющим фактором, обеспечивающим экономическую эффективность решения задач на ЭВМ, так как для задач, решение которых осуществляется один раз, целесообразность использования ЭВМ, как правило, диктуется внеэкономическими категориями.

Результатность алгоритма - свойство, означающее, что для любых допустимых исходных данных он должен через конечное число шагов (или итераций) завершить работу.

Дискретность алгоритма - свойство, означающее возможность разбиения определенного алгоритмического процесса на отдельные элементарные действия.

Таким образом, алгоритм дает возможность чисто механически решать любую задачу из некоторого класса однотипных задач.

Составление (адаптация) программ (кодирование) является завершающим этапом технологического процесса разработки программных средств. Он предшествует началу непосредственно машинной реализации алгоритма решения задачи. Процесс кодирования заключается в переводе описания алгоритма на один из доступных для ЭВМ языков программирования. В процессе составления программы для ЭВМ конкретизируются тип и структура используемых данных, а последовательность действий, реализующих алгоритм, отражается посредством конкретного языка программирования.

Этап тестирования и отладки. Оба эти процесса функционально связаны между собой, хотя их цели несколько отличаются друг от друга.

Тестирование представляет собой совокупность действий, назначенных для демонстрации правильности работы программы в заданных диапазонах изменения внешних условий и режимов эксплуатации программы. Цель тестирования заключается в демонстрации отсутствия (или выявлении) ошибок в разработанных программах на заранее подготовленном наборе контрольных примеров.

Процессу тестирования сопутствует понятие "отладка", которое подразумевает совокупность действий, направленных на устранение ошибок в программах, начиная с момента обнаружения фактов ошибочной работы программы и завершая устранением причин их возникновения.

По своему характеру (причине возникновения) ошибки в программах делятся на синтаксические и логические.

Синтаксические ошибки в программе представляют собой некорректную запись отдельных языковых конструкций с точки зрения правил их представления для выбранного языка программирования. (ошибки выявляются автоматически)

Далее проверяется логика работы программы на исходных данных. При этом возможны следующие основные формы проявления логических ошибок:

- в какой-то момент программа не может продолжать работу (возникает программное прерывание, обычно сопровождающееся указанием места в программе, где оно произошло);
- программа работает, но не выдает всех запланированных результатов и не выходит на останов (происходит ее "зацикливание");
- программа выдает результаты и завершает свою работу, но они полностью или частично не совпадают с контрольными.

После выявления логических ошибок и устранения причин их возникновения в программу вносятся соответствующие исправления и отладка продолжается.

Программа считается отлаженной, если она безошибочно выполняется на достаточно представительном наборе тестовых данных, обеспечивающих проверку всех ее участков (ветвей).

Процесс тестирования и отладки программ имеет итерационный характер и считается одним из наиболее трудоемких этапов процесса разработки программ. По оценкам специалистов, он может составлять от 30 до 50% в общей структуре затрат времени на разработку проектов и зависит от объема и логической сложности разрабатываемых программных комплексов.

Для сокращения затрат на проведение тестирования и отладки в настоящее время широко применяются специальные программные средства тестирования (например, генераторы тестовых данных) и приемы отладки (например, метод трассировки программ, позволяющий выявлять, все ли ветви программы были задействованы при решении задачи с заданными наборами исходных данных).

После завершения процесса тестирования и отладки программные средства вместе с сопроводительной документацией передаются пользователю *для эксплуатации*.

Основное назначение сопроводительной документации - обеспечить пользователя необходимыми инструктивными материалами по работе с программными средствами. Состав сопроводительной документации обычно оговаривается заказчиком (пользователем) и разработчиком на этапе подготовки технического задания на программное средство. Как правило, это документы, регламентирующие работу пользователя в процессе эксплуатации программы, а также содержащие информацию о программе, необходимую в случае возникновения потребности внесения изменений и дополнений в нее.

При передаче пользователю разработанных прикладных программных средств создается специальная комиссия, включающая в свой состав представителей разработчиков и заказчиков (пользователей). Комиссия в соответствии с заранее составленным и утвержденным обеими сторонами планом проводит работы *по приемке-передаче программных средств* и сопроводительной документации. По завершении работы комиссии оформляется акт приемки-передачи.

В процессе внедрения и эксплуатации прикладных программных средств могут выявляться различного рода ошибки, не обнаруженные разработчиком при тестировании и отладке программных средств. Поэтому при реализации достаточно сложных и ответственных программных комплексов по согласованию пользователя (заказчика) с разработчиком *этап эксплуатации* программных средств может быть разбит на два подэтапа: *экспериментальная (опытная) и промышленная эксплуатация*.

Смысл экспериментальной эксплуатации заключается во внедрении разработанных программных средств на объекте заказчика с целью проверки их работоспособности и удобства работы пользователей при решении реальных задач в течение достаточно длительного периода времени (обычно не менее года). Только после завершения периода экспериментальной эксплуатации и устранения выявленных при этом ошибок и учета замечаний программное средство передается в промышленную эксплуатацию.

Для повышения качества работ, оперативности исправления ошибок, выявляемых в процессе эксплуатации программных средств, также выполнения различного рода модификаций, в которых может возникнуть необходимость в ходе эксплуатации, разработчик может по договоренности с пользователем осуществлять их сопровождение.

Описанная схема технологического процесса разработки прикладных программных средств отражает их "жизненный цикл", т.е. временной интервал с момента зарождения программы до момента полного отказа от ее эксплуатации.

7.2 Современные методы и средства разработки программного обеспечения

7.1.1 Современные методы разработки ПО

Метод нисходящего проектирования (метод пошаговой детализации, метод иерархического проектирования, top-down-подход)

Суть метода заключается в определении спецификаций компонентов системы путем последовательного выделения в ее составе отдельных составляющих и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать и реализовывать.

Этот метод является незаменимым при разработке сложных по характеру и больших по объему программ, когда к их разработке необходимо привлекать большое число программистов, работающих параллельно. Он позволяет концентрировать внимание разработчиков на наиболее ответственных частях программы, а также облегчает возможность постоянного контроля за ее работоспособностью по мере разработки, отладки и объединения отдельных составляющих программ за счет организации непрерывности этого процесса в течение всей разработки.

Для ускорения разработки программного комплекса часто вместо некоторых программ нижнего уровня, находящихся в процессе разработки, могут применяться специальные "программы-заглушки". Программы-заглушки требуются только на ранних стадиях разработки для того, чтобы не сдерживать общий ход создания программного комплекса. Суть программы-заглушки заключается в том, что при обращении к ней в соответствии с заданным набором исходных тестовых данных она не формирует, а выбирает результат "решения" из заранее подготовленного набора. Благодаря этому обеспечивается возможность имитировать работу на ЭВМ реально создаваемой программы, а следовательно, осуществлять проверку работоспособности программ верхнего уровня еще до того, как будут разработаны и отлажены все составляющие программы нижнего уровня.

Модульное проектирование

Реализация метода *нисходящего проектирования* тесно связана с другим понятием программирования - *модульным проектированием*, так как

на практике при декомпозиции сложной программы возникает вопрос о разумном пределе ее дробления на составные части. Вместе с тем понятие модульности нельзя сводить только к представлению сложных программных комплексов в виде набора отдельных функциональных блоков.

Модуль - это последовательность логически взаимосвязанных фрагментов задачи, оформленных как отдельная часть программы. При этом программные модули должны обладать следующими свойствами:

- на модуль можно ссылаться (т.е. обращаться к нему) по имени, в том числе и из других модулей;
- по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;
- модуль должен иметь один вход и выход;
- модуль должен иметь небольшой размер, обеспечивающий его обзорность.

При разработке сложных программ в них выделяют головной управляющий модуль, подчиненные ему модули, обеспечивающие реализацию отдельных функций управления, функциональную обработку (т.е. непосредственную реализацию основного назначения программного комплекса), а также вспомогательные модули, обеспечивающие сервисное обслуживание пакета (например, сбор и анализ статистики работы программы, обработка различного рода ошибочных ситуаций, обучение и выдача подсказок и т.п.).

Модульный принцип разработки программ обладает следующими преимуществами:

- большую программу могут разрабатывать одновременно несколько исполнителей, и это позволяет сократить сроки ее разработки;
- появляется возможность создавать и многократно использовать в дальнейшем библиотеки наиболее употребимых программ;
- упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;
- возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем для контроля за ходом исполнения программ;
- обеспечивается более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.

Преимущества модульного принципа построения программ особенно наглядно проявляются на этапе сопровождения и модификации программных продуктов, позволяя значительно сократить затраты сил и средств на реализацию этого этапа.

Структурное программирование

Актуальная для начального периода развития и использования ЭВМ проблема разработки программ, занимающих минимум основной памяти и выполняющихся за кратчайшее время, в последующем в связи резким падением стоимости аппаратной части ЭВМ, значительным возрастанием их быстродействия и объемов памяти сменилась необходимостью разработки и

применения принципиально новых методов составления программ. Все это нашло свое воплощение в разработке принципа *структурного программирования*. Одной из целей структурного программирования было стремление облегчить разработку и отладку программных модулей, а главное - их последующее сопровождение и модификацию.

В настоящее время структурное программирование - это целая дисциплина, объединяющая несколько взаимосвязанных способов создания ясных, легких для понимания программ. Эффективность применения современных универсальных языков программирования во многом определяется удобством написания с их помощью структурных программ.

CASE-технологии

За последнее десятилетие в области средств автоматизации программирования сформировалось новое направление под общим названием CASE-технологии (Computer Aided Software Engineering).

CASE-технология представляет собой совокупность средств системного анализа, проектирования, разработки и сопровождения сложных программных систем, поддерживаемых комплексом взаимоувязанных инструментальных средств автоматизации всех этапов разработки программ. Благодаря структурным методам CASE-технология на стадиях анализа и проектирования обеспечивает разработчиков широкими возможностями для различного рода моделирования, а централизованное хранение всей необходимой для проектирования информации и контроль за целостностью данных гарантируют согласованность взаимодействия всех специалистов, занятых в разработке ПО.

Технологии RAD

В начале 80-х годов появилась методология, по которой разработка программы начиналась не после завершения процесса выработки окончательных требований к ней, а как только устанавливались требования на первый, "стартовый" (пилотный) вариант прикладной программы, позволяющий начать содержательную работу по ее реализации на компьютере.

Это дало пользователю возможность, получая уже с первых шагов конкретное представление о характере реализации задачи, уточнять ее постановку. Тем самым облегчался процесс экспериментального поиска нужного решения автоматизации задачи. Благодаря тесному взаимодействию разработчика с заказчиком (пользователем) на самом ответственном этапе создания прикладных программ между ними достигалось быстрое взаимопонимание цели поставленной задачи и возможности ее автоматизации в данных конкретных условиях. Это повышало скорость разработки программ и послужило основанием для названия такой технологии *RAD (Rapid Application Development* - быстрая разработка программ), которая получила широкое распространение.

Data Warehouse

Другое направление разработки прикладных программных средств, олицетворяющее собой современный подход к реализации широкого круга

задач для принятия управленческих решений, базируется на концепции создания специального хранилища данных (Data Warehouse). Основное отличие концепции Data Warehouse от традиционного представления баз данных заключается в следующем:

- во-первых, в том, что актуализация данных в Data Warehouse означает не обновление элементов информации, а добавление новых элементов к уже имеющимся (что расширяет возможности проведения различного рода сравнительного анализа);
- во-вторых, в том, что наряду с информацией, непосредственно отражающей состояние системы управления, в Data Warehouse аккумулируются и метаданные.

Метаданные (данные о данных) облегчают возможность визуального представления содержимого Data Warehouse, позволяют, "перемещаясь" по хранилищу, быстро отбирать необходимые данные для последующей обработки.

Основные типы метаданных Data Warehouse отражают:

- структуру и содержимое хранилища;
- соответствие между исходными и выходными данными;
- объемные характеристики данных;
- критерии архивирования;
- отношения между данными;
- информацию по кодированию;
- интервал жизни данных и т.п.

Концепция *Data Warehouse* поддерживается RAD средствами разработки прикладного ПО.

Концепция Data Warehouse обеспечивает возможность разработки программных приложений для поддержки процессов принятия решений с использованием OLAP-систем.

Система OLAP (On-Line Analytical Process) предоставляет возможность разработки информационных систем, ориентированных на уна организацию многомерных баз данных и создание корпоративных сетей, а также обеспечивает поддержку Web-технологий в сетях Internet/Intranet

Успешное применение инструментальных средств OLAP-систем объясняется быстротой разработки приложений, гибкостью и широкими возможностями в области доступа к данным и их преобразования. В настоящее время на рынке ПО предлагается большое число OLAP-стем, разработчиками которых являются различные фирмы, например IBM, Informix, Microsoft, Oracle, Sybase и др.

7.1.2 Инструментарий технологии программирования

Инструментарий технологии программирования - программные продукты поддержки (обеспечения) технологии программирования.

В рамках этого направления сформировались следующие группы программных продуктов (рис. 7.2):

- 1) средства для создания приложений, включающие:
 - локальные средства, обеспечивающие выполнение отдельных работ по созданию программ;
 - интегрированные среды разработчиков программ, обеспечивающие выполнение комплекса взаимосвязанных работ по созданию программ;
- 2) средства для создания информационных систем (CASE-технология), представляющие методы анализа, проектирования и создания программных систем и предназначенные для автоматизации процессов разработки и реализации информационных систем.

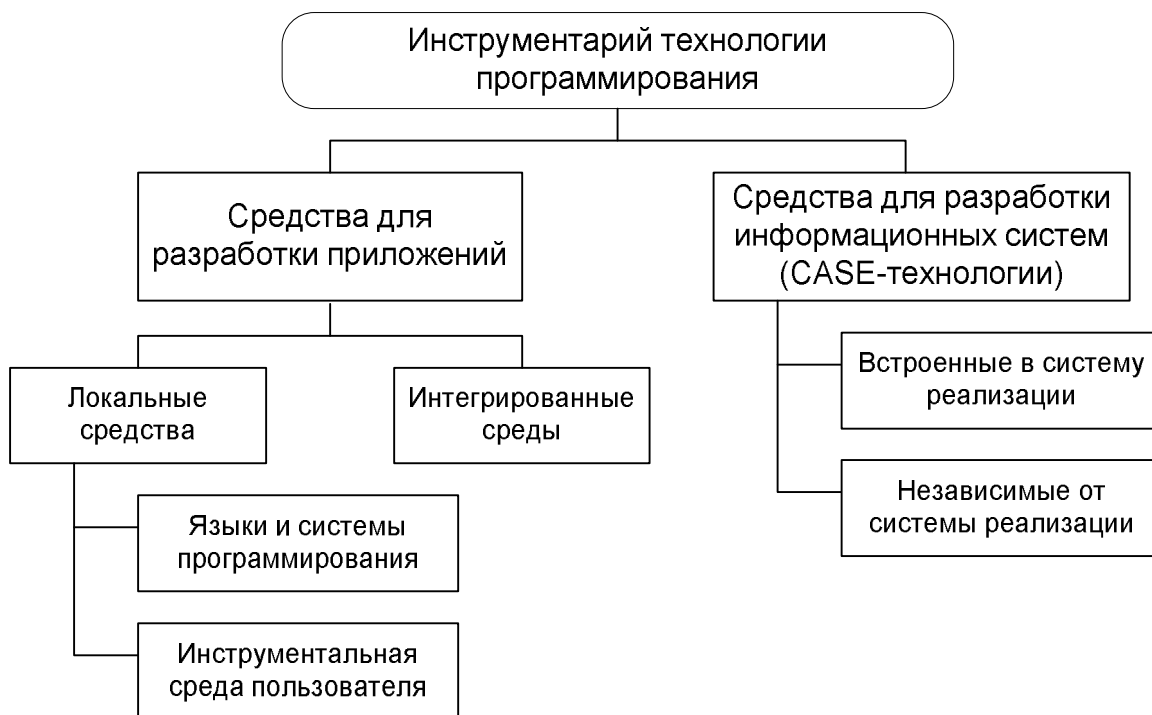


Рис. 7.2 - Классификация инструментария технологии программирования

7.1.3 Средства для создания приложений

Локальные средства разработки программ

Эти средства на рынке программных продуктов наиболее представительны и включают языки и системы программирования, а также инструментальную среду пользователя.

Язык программирования - формализованный язык для описания алгоритма решения задачи на компьютере.

Средства для создания приложений - совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.

Языки программирования можно условно разделить на следующие классы (если в качестве признака классификации взять синтаксис образования конструкций языка):

- машинные языки (computer language) - языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды);
- машинно-ориентированные языки (computer-oriented language) - языки программирования, которые отражают структуру конкретного типа компьютера (ассемблеры);
- алгоритмические языки (algorithmic language) - языки программирования, не зависящие от архитектуры компьютера (Паскаль, Си, Фортран, Бейсик и др.);
- процедурно-ориентированные языки (procedure-oriented language) - языки программирования, где имеется возможность написания программы как совокупности процедур (подпрограмм);
- проблемно-ориентированные языки (universal programming language) - языки программирования, предназначенные для решения задач определенного класса (Лисп, Пролог, Симула и др.);
- интегрированные системы программирования.

Другой классификацией языков программирования является их деление на языки, ориентированные на реализацию основ структурного программирования, и объектно-ориентированные языки, поддерживающие понятие объектов и их свойств и методов обработки.

Программа, подготовленная на языке программирования, проходит этап трансляции, когда происходит преобразование исходного кода программы (source code) в объектный код (object code), который далее пригоден к обработке редактором связей. Редактор связей специальная программа, обеспечивающая построение загрузочного модуля (load module), пригодного к выполнению (рис. 7.3).

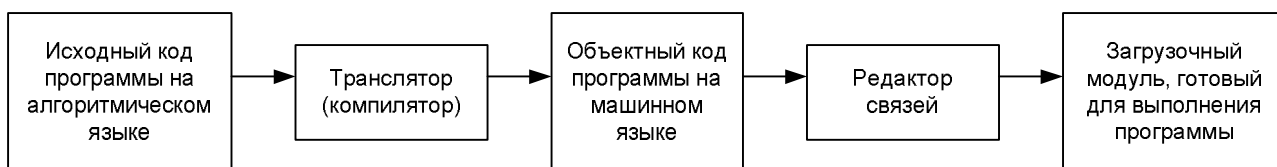


Рис. 7.3 - Схема процесса создания загрузочного модуля программы

Трансляция может выполняться с использованием средств компиляторов (compiler) или интерпретаторов (interpreter). Компиляторы транслируют всю программу, но без ее выполнения. Интерпретаторы, в

отличие от компиляторов, выполняют пооператорную обработку и выполнение программы.

Существуют специальные программы, предназначенные для трассировки и анализа выполнения программ, так называемые отладчики (debugger). Лучшие отладчики позволяют осуществить трассировку (отслеживание выполнения программы в пооператорном варианте), идентификацию места и вида ошибок в программе, наблюдение за изменением значений переменных, выражений и т.п. Для отладки и тестирования правильности работы программ создается база данных контрольного примера.

Более мощным средством разработки программ являются *системы программирования*.

Системы программирования (programming system) включают:

- компилятор;
- интегрированную среду разработчика программ;
- отладчик;
- средства оптимизации кода программ;
- набор библиотек (возможно с исходными текстами программ);
- редактор связей;
- сервисные средства (утилиты) для работы с библиотеками текстовыми и двоичными файлами;
- справочные системы;
- документатор исходного кода программы;
- систему поддержки и управления проектом программного комплекса.

Средства поддержки проектов - новый класс средств разработки программного обеспечения, предназначенный для:

- отслеживания изменений, выполненных разработчиками программ;
- поддержки версий программы с автоматической разnosкой изменений;
- получения статистики о ходе работ проекта.

Инструментальная среда пользователя представлена специальными средствами, встроенными в пакеты прикладных программ, такими, как:

- библиотека функций, процедур, объектов и методов обработки;
- макрокоманды;
- клавишные макросы; языковые макросы;
- программные модули-вставки; конструкторы экранных форм и отчетов;
- генераторы приложений; языки запросов высокого уровня;
- языки манипулирования данными; конструкторы меню и многое другое.

Средства отладки и тестирования программ предназначены для подготовки разработанной программы к промышленной эксплуатации.

Интегрированные среды разработки программ

Дальнейшим развитием локальных средств разработки программ, являются интегрированные программные среды разработчиков.

Основное назначение инструментария данного вида - повышение производительности труда программистов, автоматизация создания кодов программ, обеспечивающих интерфейс пользователя графического типа, разработка приложений для архитектуры клиент-сервер, запросов и отчетов.

7.1.4 CASE-технологии

CASE-технологии - относительно новое направление, сформировавшееся на рубеже 80-х годов.

CASE-технологии делятся на две группы:

- встроенные в систему реализации, в которых все решения по проектированию и реализации привязаны к выбранной системе являющейся базами данных (СУБД);
- независимые от системы реализации, в которых все решения по проектированию ориентированы на унификацию начальных этапов жизненного цикла, средств их документирования и обеспечивают большую гибкость в выборе средств реализации.

Основное достоинство CASE-технологии - поддержка коллективной работы над проектом за счет возможности работы в локальной сети разработчиков, экспорта/импорта любых фрагментов проекта, организационного управления проектом.

Некоторые CASE-технологии ориентированы только на системных проектировщиков и предоставляют специальные графические средства для изображения различного вида моделей:

- диаграмм потоков данных (DFD - data flow diagrams) совместно со словарями данных и спецификациями процессов;
- диаграмм "сущность-связь" (ERD - entity relationship diagrams), являющихся информационной моделью предметной области;
- диаграмм переходов состояний (STD - state transition diagrams), учитывающих события и реакцию на них системы обработки данных.

Диаграммы DFD устанавливают связь источников информации с потребителями, выделяют логические функции (процессы) образования информации, определяют группы элементов данных и их хранилища (базы данных).

Описание структуры потоков данных, определение их компонентов хранятся в актуальном состоянии в словаре данных, который выступает как база данных проекта. Каждая логическая функция может детализироваться с помощью DFD нижнего уровня согласно методам исходящего проектирования.

Этими CASE-технологиями выполняются автоматизированное проектирование спецификаций программ (задание основных характеристик для разработки программ) и ведение словаря данных.

Другой класс CASE-технологий поддерживает только разработку программ, включая:

- автоматическую генерацию кодов программ на основании их спецификаций;
- проверку корректности описания моделей данных и схем потоков данных;
- документирование программ согласно принятым стандартам и актуальному состоянию проекта;
- тестирование и отладку программ.

Кодогенерация программ выполняется двумя способами: создание каркаса программ и создание полного продукта. Каркас программы служит для последующего ручного варианта редактирования исходных текстов, обеспечивая возможность вмешательства программиста; полный продукт не редактируется вручную.

В рамках CASE-технологий проект сопровождается целиком, а не только его программные коды. Проектные материалы, подготовленные в CASE-технологии, служат заданием программистам, а само программирование скорее сводится к кодированию - переводу на определенный язык структур данных и методов их обработки, если не предусмотрена автоматическая кодогенерация.

7.3 Языки и системы программирования

7.3.1 Развитие языков программирования

Поколения языков программирования

Языки программирования принято делить на пять поколений.

В первое поколение входят языки, созданные в начале 50-х годов, когда только появились первые компьютеры. Это был первый язык ассемблера, созданный по принципу "одна инструкция - одна строка".

Расцвет *второго поколения* языков программирования пришелся на конец 50-х - начало 60-х годов. Тогда был разработан символический ассемблер, в котором появилось понятие переменной. Он стал первым полноценным языком программирования. Благодаря его возникновению заметно возросли скорость разработки и надежность программ.

Появление *третьего поколения* языков программирования принято относить к 60-м годам. В это время возникли универсальные языки высокого уровня, с их помощью удается решать задачи из любых областей. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций, позволили резко повысить производительность труда программистов. Понятная большинству пользователей структура этих языков привлекла к написанию небольших программ (как правило, инженерного или экономического характера) Подавляющее большинство языков этого поколения успешно применяется и сегодня.

С начала 70-х годов по настоящее время продолжается период языков *четвертого поколения*. Эти языки предназначены для реализации крупных проектов, повышения их надежности и скорости создания. Они ориентированы на специализированные области применения, где хороших результатов можно добиться, используя не универсальные, а проблемно-ориентированные языки, оперирующие конкретными понятиями узкой предметной области.

Как правило, в эти языки встраиваются мощные операторы, позволяющие одной строкой описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк исходного кода.

Рождение языков *пятого поколения* произошло в середине 90-х годов. К ним относятся также системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования.

Главная идея, которая закладывается в эти языки, - возможность автоматического формирования результирующего текста на универсальных языках программирования (который потом требуется откомпилировать). Инструкции же вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

Обзор языков программирования высокого уровня

Fortran (Фортран)

Это первый компилируемый язык, созданный в 50-е годы.

Программисты, разрабатывавшие программы исключительно на ассемблере, выражали серьезное сомнение в возможности появления высокопроизводительного языка высокого уровня, поэтому основным критерием при разработке компиляторов Фортрана являлась эффективность исполняемого кода. Хотя в Фортране впервые был реализован ряд важнейших понятий программирования, удобство создания программ было принесено в жертву возможности получения эффективного машинного кода. Однако для этого языка было создано огромное количество библиотек, начиная от статистических комплексов и заканчивая пакетами управления спутниками. Фортран продолжает активно использоваться во многих организациях. Имеется стандартная версия Фортрана HPF (High Performance Fortran) для параллельных суперкомпьютеров со множеством процессоров.

Cobol (Кобол)

Это компилируемый язык для применения в экономической области и решения бизнес - задач, разработанный в начале 60-х годов. Он отличается большой "многословностью" – его операторы иногда выглядят как обычные английские фразы. В Коболе были реализованы очень мощные средства работы с большими объемами данных, хранящимися на различных внешних носителях. На этом языке создано очень много приложений, которые эксплуатируются и сегодня.

Algol (Алгол). Компилируемый язык, созданный в 1960г. Он был призван заменить Фортран, но из-за более сложной структуры не получил широкого распространения. В 1968г. была создана версия Алгол 68, по своим возможностям и сегодня опережающая многие языки программирования, однако из-за отсутствия достаточно эффективных компьютеров для нее не удалось своевременно создать хорошие компиляторы.

Pascal (Паскаль)

Язык Паскаль, созданный в конце 70-х годов, во многом напоминает Алгол, но в нем ужесточен ряд требований к структуре программы и имеются возможности, позволяющие успешно применять его при создании крупных проектов.

Basic (Бейсик)

Для этого языка имеются и компиляторы, и интерпретаторы, а по популярности он занимает первое место в мире. Он создавался в 60-х годах в качестве учебного языка и очень прост в изучении.

C (Си)

Данный язык был создан в лаборатории Bell и первоначально не рассматривался как массовый. Он планировался для замены ассемблера, чтобы иметь возможность создавать столь же эффективные и компактные программы, и в то же время не зависеть от конкретного типа процессора. Си во многом похож на Паскаль и имеет дополнительные средства для прямой работы с памятью (указатели). На этом языке в 70-е годы написано множество прикладных и системных программ и ряд известных операционных систем (Unix).

C++ (Си++)

Си++ - это объектно-ориентированное расширение языка Си, разработан в 1980 г. В нем реализовано множество новых мощных возможностей, которые позволили резко повысить производительность труда программистов, однако создание сложных и надежных программ требует от разработчиков профессиональной подготовки высокого уровня.

Java (Ява)

Этот язык был создан компанией Sun в начале 90-х годов на основе Си++. Он призван упростить разработку приложений на основе Си++ путем исключения из него всех низкоуровневых возможностей. Но главная особенность этого языка - компиляция не в машинный код, а в платформенно-независимый байт-код (каждая команда занимает один байт). Этот байт-код может выполняться с помощью интерпретатора - виртуальной Java-машины JVM (Java Virtual Machine), версии которой созданы сегодня для любых платформ.

Особое внимание в развитии этого языка уделяется двум направлениям:

- поддержке всевозможных мобильных устройств и микрокомпьютеров, встраиваемых в бытовую технику (технология Jini);

- созданию платформно - независимых программных модулей, способных работать на серверах в глобальных и локальных сетях с различными операционными системами (технология Java Beans).

Пока недостаток этого языка - невысокое быстродействие.

Языки программирования баз данных

Эта группа языков отличается от алгоритмических языков, прежде всего решаемыми задачами. База данных - это файл (или группа файлов), представляющий собой упорядоченный набор записей, имеющих единообразную структуру и организованных по единому шаблону (как правило, в табличном виде). База данных может состоять из нескольких таблиц. Удобно хранить в базах данных различные сведения из справочников, картотек, журналов бухгалтерского учета и т. д. При работе с базами данных чаще всего требуется выполнять следующие операции:

Первые базы данных появились очень давно, как только появилась потребность в обработке больших массивов информации и выборки групп записей по определенным признакам. Для этого был создан структурированный язык запросов SQL (Structured Query Language). Он основан на мощной математической теории и позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а группами записей.

Для управления большими базами данных и их эффективной обработки разработаны СУБД (Системы Управления Базами Данных).

Практически фактически в каждой СУБД помимо поддержки языка SQL имеется свой уникальный язык, ориентированный на особенности этой СУБД и не переносимый на другие системы.

Сегодня в мире насчитывается пять ведущих производителей СУБД: Microsoft (SQL Server), IBM (DB2), Oracle, Software AG (Adabas), Informix и Sybase. Их продукты нацелены на поддержку одновременной работы тысяч пользователей в сети, а базы данных могут храниться в распределенном виде на нескольких серверах.

С появлением персональных компьютеров были созданы так называемые настольные СУБД. Родоначальником современных языков программирования баз данных для ПК принято считать СУБД dBase II, язык которой был интерпретируемым. Затем для него были созданы компиляторы, появились СУБД FoxPro и Clipper, поддерживающие диалекты этого языка. Сегодня похожие, но несовместимые версии языков семейства dBase реализованы в продуктах Visual FoxPro фирмы Microsoft и Visual dBase фирмы Inprise.

Языки программирования для Интернета

С активным развитием глобальной сети было создано немало популярных языков программирования, адаптированных специально для Интернета.

Все они отличаются характерными особенностями: языки являются интерпретируемыми, интерпретаторы для них распространяются бесплатно, а сами программы - в исходных текстах. Такие языки называют скрипт-языками.

HTML. Общеизвестный язык для оформления документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц. Все Web-страницы написаны на языке HTML или используют его расширения.

Perl. В 80-х годах Ларри Уолл разработал язык Perl. Он задумывался как средство эффективной обработки больших текстовых файлов, генерации текстовых отчетов и управления задачами. По мощности Perl значительно превосходит языки типа Си. В него введено много часто используемых функций работы со строками, массивами, всевозможные средства преобразования данных, управления процессами, работы с системной информацией и др.

VRML. В 1994г. был создан язык VRML для организации виртуальных трехмерных интерфейсов в Интернете. Он позволяет описывать в текстовом виде различные трехмерные сцены, освещение и тени, текстуры (покрытия объектов), вращать в любых направлениях, масштабировать, регулировать освещенность и т. д.

Языки моделирования

При создании программ и формировании структур баз данных нередко применяются формальные способы их представления - формальные нотации, с помощью которых можно визуально представить таблицы баз данных, поля, объекты программы и взаимосвязи между ними в системе, имеющей специализированный редактор и генератор исходных текстов программ на основе созданной модели. Такие системы называются CASE-системами. В них активно применяются нотации IDEF, а в последнее время все большее распространение получает UML.

7.3.2 Современные системы программирования

Основы визуального программирования интерфейса

Один из тупиков или кризисов программирования, не так давно был связан с разработкой графического интерфейса пользователя. Программирование вручную всяких привычных пользователю окон, кнопок, меню, обработка событий мыши и клавиатуры, включение в программы изображений и звука требовало все больше и больше времени программиста. В ряде случаев весь этот сервис начинал занимать до 80-90% объема программных кодов. Причем весь этот труд нередко пропадал почти впустую, поскольку через год – другой менялся общепринятый стиль графического интерфейса и все приходилось начинать заново.

Выход из этой ситуации обозначился благодаря двум подходам.

Первый - стандартизация многих функций интерфейса, благодаря чему появилась возможность использовать библиотеки, имеющиеся, например, в Windows. В итоге при смене стиля графического интерфейса (например, при переходе от Windows 3.x к Windows 95) приложения смогли автоматически приспосабливаться к новой системе без какого-либо перепрограммирования.

Вторым революционным шагом явилось появление визуального программирования, возникшего в Visual Basic и нашедшего блестящее воплощение в Delphi и C++Builder фирмы Borland.

Визуальное программирование позволило свести проектирование пользовательского интерфейса к простым и наглядным процедурам, которые дают возможность за минуты или часы сделать то, на что ранее уходили месяцы работы.

Пример. В Delphi это выглядит следующим образом. Вы работаете в Интегрированной Среде Разработки Delphi, которая предоставляет Вам формы (в приложении их может быть несколько) на которых умещаются компоненты. Обычно это оконная форма, хотя могут быть и невидимые формы. На форму с помощью мыши переносятся и размещаются пиктограммы компонентов, имеющих в библиотеках Delphi. С помощью простых манипуляций можно изменять размеры расположение этих компонентов. При этом все время в процессе проектирования видно результат - изображение формы и расположенных на ней компонентов. Результаты проектирования видно, даже не компилируя программу.

Основное достоинство заключается в том, что во время проектирования формы и размещения на ней компонентов, Delphi автоматически формирует коды программы, включая в нее соответствующие фрагменты, описывающие данный компонент. А затем в соответствующих диалоговых окнах пользователь может изменить заданные по умолчанию значения компонентов и, при необходимости написать обработчики каких-то событий. Фактически, проектирование сводится к размещению компонентов на форме, заданию некоторых их свойств и написанию, при необходимости, обработчиков событий.

Компоненты могут быть визуальные, видимые при работе приложения, и не визуальные, выполняющие те или иные служебные функции. Визуальные компоненты сразу видны на экране в процессе проектирования в таком же виде, в каком их увидит пользователь во время выполнения приложения.

В библиотеки визуальных компонентов Delphi включено множество типов компонентов и их номенклатура очень быстро расширяется от версии к версии. Имеющегося уже сейчас вполне достаточно, чтобы построить практически любое самое замысловатое приложение, не прибегая к созданию новых компонентов. При этом даже неопытный программист, может создавать приложения, которые выглядят профессионально.

Основные системы программирования

Из универсальных языков программирования сегодня наиболее популярны следующие: Бейсик (Basic), Паскаль (Pascal), Си++ (C++), Ява (Java).

Для каждого из этих языков программирования сегодня имеется • немало систем программирования, выпускаемых различными фирмами и ориентированных на различные модели ПК и операционные системы.

Наиболее популярны следующие визуальные среды быстрого проектирования программ для Windows:

- Basic: Microsoft Visual Basic;
- Pascal: Borland Delphi;
- C++: Borland C++Builder;
- Java: Symantec Cafe.

Для разработки серверных и распределенных приложений можно использовать систему программирования Microsoft Visual C++, продукты фирмы Borland, практически любые средства программирования на Java.

7.4 Архитектура программных систем

В то время как большинство автономных приложений - офисные программы, среды разработки, системы подготовки текстов и изображений - выполняются на одном компьютере, крупные информационные комплексы (например, система автоматизации предприятия) состоят из десятков и сотен отдельных программ, которые взаимодействуют друг с другом по сети, выполняясь на разных компьютерах. В таких случаях говорят, что они работают в различной программной архитектуре.

- 1) **Автономные приложения.** Работают на одном компьютере.
- 2) **Приложения в файл-серверной архитектуре.** Компьютеры пользователей системы объединены в сеть, при этом на каждом из них (на клиентском месте) запущены копии одной и той же программы, которые обращаются за данными к серверу, который хранит файлы, одновременно доступные всем пользователям (как правило, это базы данных). Сервер обладает повышенной надежностью, высоким быстродействием, большим объемом памяти, на нем установлена специальная серверная версия операционной системы. При одновременном обращении нескольких программ к одному файлу, например, с целью его обновления, могут возникнуть проблемы, связанные с неоднозначностью определения его содержимого. Поэтому каждое изменение общедоступного файла выделяется в транзакцию (элементарную операцию по обработке данных, имеющую фиксированное начало, конец (успешное или неуспешное завершение) и ряд других характеристик). Особенность этой архитектуры в том, что все вычисления выполняются на клиентских местах, что требует наличия на них достаточно производительных ПК (это так называемые системы с толстым клиентом - программой, которая выполняет всю обработку получаемой от сервера информации).

3) **Приложения в клиент-серверной архитектуре.** Эта архитектура похожа на предыдущую, только сервер помимо простого обеспечения одновременного доступа к данным, способен еще выполнять программы, которые берут на себя определенный объем вычислений (в файл-серверной архитектуре он реализуется полностью на клиентских установках). Благодаря этому удастся повысить общую надежность системы, так как сервер работает значительно более устойчиво, чем ПК, и снять лишнюю нагрузку с клиентских мест, на которых удастся использовать. Запускаемые на них приложения осуществляют небольшие объемы вычислений, а иногда занимаются только отображением получаемой от сервера информации, поэтому они называются тонкими клиентами.

4) **Приложения в многозвенной архитектуре.** Недостаток предыдущей архитектуры в том, что резко возрастает нагрузка на сервер, а если он выходит из строя, то работа всей системы останавливается. Поэтому в систему добавляется так называемый сервер приложений, на котором выполняется вся вычислительная работа. Другой сервер баз данных обрабатывает запросы пользователей, на третьем может быть установлена специальная программа - монитор транзакций, которая оптимизирует обработку транзакций и балансирует нагрузку на серверы. В большинстве практических случаев все серверы соединены последовательно, и выход из строя одного звена если и не останавливает всю работу, то по крайней мере, снижает производительность системы.

5) **Приложения в распределенной архитектуре.** Чтобы избежать недостатков рассмотренных архитектур, были придуманы специальные технологии, позволяющие создавать программу в виде набора компонентов, которые можно запускать на любых серверах, связанных сеть (компоненты как бы распределены по сети). Основное преимущество подобного подхода в том, что при выходе из строя любого компьютера специальные программы-мониторы, которые следят за корректностью работы, сразу перезапускают временно пропавший компонент на другом компьютере. При этом общая надежность всей системы становится очень высокой, а вычислительная нагрузка распределяется между серверами оптимальным образом. Доступ к возможностям любого компонента, предназначенного для общения с пользователем, осуществляется с произвольного клиентского места. Так как все вычисления происходят на серверах, появляется возможность создавать сверхтонкие клиенты - программы только отображающие получаемую из сети информацию и требующие минимальных компьютерных ресурсов. Благодаря этому доступ к компонентной системе возможен не только с ПК, но и с небольших мобильных устройств. Частный случай компонентного подхода - доступ к серверным приложениям из браузеров через Интернет.

Сегодня наиболее популярны три компонентные технологии: CORBA консорциума OMG, Java Beans компании Sun, COM+ корпорации Microsoft.

Эти технологии будут определять развитие информационной индустрии в ближайшие десятилетия.